



# 存储资源盘活系统

用户手册 - 管理 API

天翼云科技有限公司

# 目 录

1 产品概述 .....	1
1.1 产品定义 .....	1
1.2 应用场景 .....	3
1.3 基本概念 .....	4
1.3.1 iSCSI .....	4
1.3.2 卷 .....	4
1.3.3 iSCSI Target .....	4
1.3.4 iSCSI 目标门户 .....	4
1.3.5 故障域 .....	5
1.3.6 数据服务 .....	5
1.3.7 监控 .....	5
1.3.8 事件 .....	5
1.3.9 日志采集 .....	5
1.3.10 告警 .....	6
1.4 术语与缩略语 .....	7
2 服务器部署要求 .....	8
2.1 环境要求 .....	8
2.2 HBlock 配置环境 - 单机版 .....	10
2.3 HBlock 配置环境 - 集群版 .....	12
3 安装访问 .....	15
3.1 安装 .....	15
3.2 访问地址 .....	17
3.3 接口 .....	18
3.3.1 URL 规则 .....	18
3.3.2 通用请求头 .....	19
3.3.3 签名方法 .....	20
3.3.4 通用响应头 .....	21
3.3.5 通用错误码 .....	22
4 管理 API .....	26
4.1 初始化 .....	26
4.1.1 初始化 .....	26
4.1.2 查询初始化进度 .....	39

4.1.3 错误码列表	41
4.2 软件许可证	48
4.2.1 导入软件许可证	48
4.2.2 批量查询软件许可证信息	53
4.2.3 查询指定软件许可证信息	59
4.2.4 错误码列表	64
4.3 卷管理	66
4.3.1 创建卷	66
4.3.2 启用卷	75
4.3.3 禁用卷	76
4.3.4 删除卷	78
4.3.5 扩容卷	79
4.3.6 修改卷	81
4.3.7 触发卷对应 Target 的主备切换（集群版适用）	84
4.3.8 批量查询卷	87
4.3.9 查询指定卷	97
4.3.10 错误码列表	104
4.4 iSCSI Target 管理	109
4.4.1 创建 iSCSI Target	109
4.4.2 修改 iSCSI Target 的 CHAP 认证	116
4.4.3 迁移 iSCSI Target（集群版适用）	119
4.4.4 修改 iSCSI Target 下每个 IQN 允许建立的最大会话数	124
4.4.5 删除 iSCSI Target	126
4.4.6 删除 CHAP	128
4.4.7 批量查询 iSCSI Target	129
4.4.8 查询指定 iSCSI Target	137
4.4.9 查询 iSCSI Target 连接	143
4.4.10 删除 iSCSI Target 连接	146
4.4.11 错误码列表	148
4.5 服务器管理	151
4.5.1 添加服务器（集群版适用）	151
4.5.2 修改服务器属性	156
4.5.3 删除服务器属性	161
4.5.4 移除服务器（集群版适用）	162
4.5.5 重启服务器上的 HBLock	166
4.5.6 批量查询服务器	168
4.5.7 查询指定服务器	179
4.5.8 添加数据目录	193

4.5.9 修改数据目录的容量配额 .....	196
4.5.10 移除数据目录 .....	199
4.5.11 错误码列表 .....	202
4.6 查询 HBlock 信息 .....	209
4.7 查询 HBlock 服务状态 .....	212
4.8 故障域（集群版适用） .....	217
4.8.1 查询故障域 .....	217
4.9 监控 .....	224
4.9.1 获取实时监控数据 .....	224
4.9.2 获取历史性能数据 .....	230
4.9.3 错误码列表 .....	239
4.10 告警 .....	240
4.10.1 批量查询 HBlock 告警信息 .....	240
4.10.2 查询指定的 HBlock 告警 .....	255
4.10.3 手动解除 HBlock 告警 .....	259
4.10.4 静默告警/解除告警静默 .....	261
4.10.5 导出告警 .....	264
4.10.6 错误码列表 .....	266
4.11 事件和日志管理 .....	267
4.11.1 查看事件 .....	267
4.11.2 导出事件 .....	272
4.11.3 发起 HBlock 日志采集 .....	275
4.11.4 批量查询采集的日志 .....	280
4.11.5 查询单个采集的日志 .....	288
4.11.6 下载采集的日志文件 .....	291
4.11.7 删除指定的采集日志 .....	293
4.11.8 删除所有的采集日志 .....	294
4.11.9 错误码列表 .....	295
4.12 HBlock 系统设置 .....	297
4.12.1 修改管理员密码 .....	297
4.12.2 邮件设置 .....	299
4.12.3 远程协助 .....	310
4.13 还原 HBlock 集群中的服务器（集群版适用） .....	318
4.13.1 还原 HBlock 集群中的服务器 .....	318
4.13.2 错误码列表 .....	321

4.14 查询 HBlock 版本 .....	322
5 附录 .....	324
5.1 转义字符表 .....	324
5.2 HBlock 服务 .....	325
5.3 用户事件列表 .....	326
5.4 系统事件列表 .....	329
5.5 监控指标 .....	332
5.6 告警列表 .....	337

# 1 产品概述

## 1.1 产品定义

HBlock 是中国电信天翼云自主研发的存储资源盘活系统（Storage Resource Reutilization System，简称 SRRS），是一款轻量级存储集群控制器，实现了全用户态的软件定义存储，将通用服务器及其管理的闲置存储资源转换成高可用的虚拟磁盘，通过标准 iSCSI 协议提供分布式块存储服务，挂载给本地服务器（或其他远程服务器）使用，实现对资源的集约利用。同时，产品拥有良好的异构设备兼容性 & 场景化适配能力，无惧 IT 架构升级带来的挑战，助力企业用户降本增效和绿色转型。

HBlock 可以像普通应用程序那样以非 root 方式安装在 Linux 操作系统中，与服务器中的其他应用混合部署，形成的高性能高可用的虚拟硬盘供业务使用。如此一来，HBlock 可以在不影响用户业务、无需额外采购设备的情况下，直接原地盘活存储资源！

传统的硬件存储阵列可以为每个逻辑卷提供低延迟和高可用性，但存在横向扩展性差、成本高的问题，并且可能形成许多“数据孤岛”，导致存储资源成本高和利用率低。传统的分布式存储虽然具有很强的吸引力，但通常存在部署复杂、性能差、稳定性差等问题。

HBlock 以完全不同的方式提供存储阵列：

- **易安装：** HBlock 安装包是一个 zip 包，可以安装在通用 64 位 x86 服务器或者 ARM 服务器上的主流 Linux 操作系统上，支持物理服务器、裸金属服务器、虚拟机。HBlock 与硬件驱动程序完全解耦，用户可以自由使用市场上最新的硬件，减少供应商锁定。
- **绿色：** HBlock 作为一组用户态进程运行，不依赖特定版本的 Linux 内核或发行版，不依赖、不修改操作系统环境，不独占整个硬盘，不干扰任何其他进程的执行。因此，HBlock 可以与其他应用同时运行在同一 Linux 操作系统实例中。我们称此功能为“绿色”。一方面，它可以帮助用户提高现有硬件资源的利用率，另一方面，它也降低了用户使用 HBlock 的门槛 — 甚至不需要虚拟机。
- **高利用率：** HBlock 支持异构硬件，集群中的每个 Linux 操作系统实例可以具有不同的硬件配置，例如不同数量的 CPU、不同的内存大小、不同容量的本地硬盘等。因此可以提高现有硬件资源的利用率。

- **高性能：** HBlock采用分布式多控架构，提供像传统硬件存储阵列一样的低延迟和高可用性，以及像传统分布式存储一样的高扩展性和高吞吐量。支持在不中断业务的情况下，从 3 台服务器扩展到数千台服务器，并从数千台服务器逐台缩小到3台服务器。
- **高质量：** 当集群中同时发生的磁盘故障数不大于逻辑卷冗余模式允许的故障数（对于3副本模式，允许的故障数为2；对于纠删码N+M模式，允许的故障数为M），不影响HBlock的数据持久性。在集群中发生单个服务器、链路或磁盘故障时，HBlock保证服务可用。HBlock是面向混沌（Chaos）环境设计的，可适用于弱网、弱算、弱盘等不确定环境，并在发布之前已经在复杂和大规模的环境中进行了充分的测试。

## 1.2 应用场景

- **边缘计算：**边缘计算是一种分布式计算范式，它使计算和数据存储更接近数据产生和操作执行的位置，而不是依赖于数千里外的数据中心，提高了响应时间，节省了带宽。边缘计算有两种存储解决方案：一是现有的边缘设备依赖本地存储单元来存储数据，这种方案存在容量有限和扩展性低的问题；二是采用企业级存储解决方案，但这种方案也存在很多问题，如标准不统一、不同硬件的接口不同，成本高昂等。**HBlock**为边缘计算提供了更好的存储方案，因为它是天生面向混沌环境而设计的，可对抗弱网、弱算、弱盘等情况，并且支持混合部署，具有可扩展性、低延迟、高可用性、灵活等特性。它可以重用边缘设备的现有存储资源，在不中断服务的情况下随时按需扩展存储容量，提供持久的数据存储。
- **第二存储：**第二存储用于存储不经常访问的非关键数据。主存储通常需要昂贵、高性能的存储系统，但第二存储可以使用经济、性能不高的设备，用来长期存储数据。第二存储的类型包括：备份和容灾数据、视频监控数据、归档数据等。如果想使用通用硬件或现有服务器来构建第二存储，**HBlock**是一个特别好的选择。**HBlock**是一个用户态进程级软件定义存储控制器，它只需要磁盘中的一个目录来存储数据，不依赖、不修改操作系统环境，不独占整个硬盘，也不干扰其他任何进程的执行。**HBlock**支持异构硬件，它允许集群中的每个Linux操作系统实例具有不同的硬件配置，例如不同数量的CPU、不同大小的内存、不同容量的本地硬盘等。因此，用户可以使用**HBlock**，将通用服务器或现有服务器构建成第二存储，节省总成本。

## 1.3 基本概念

### 1.3.1 iSCSI

iSCSI (Internet Small Computer System Interface, 互联网小型计算机系统接口) 是一种基于 TCP/IP 及 SCSI-3 协议下的存储技术, 用来建立和管理 IP 存储设备、主机和客户机等之间的相互连接, 并创建存储区域网络 (SAN)。

### 1.3.2 卷

LUN (Logical Unit Number, 卷) 是逻辑单元号, 用于标识逻辑单元的数字。

最小副本数: 对于副本模式的卷, 假设卷副本数为  $X$ , 最小副本数为  $Y$  ( $Y$  必须  $\leq X$ ), 该卷每次写入时, 至少  $Y$  份数据写入成功, 才视为本次写入成功。对于 EC  $N+M$  模式的卷, 假设该卷最小副本数设置为  $Y$  (必须满足  $N \leq Y \leq N+M$ ), 必须满足总和至少为  $Y$  的数据块和校验块写入成功, 才视为本次写入成功。

### 1.3.3 iSCSI Target

iSCSI Target 是位于 iSCSI 服务器上的存储资源。它是一个通过 IP 网络基础设施来连接数据存储设备的协议, 允许将远程存储设备映射到本地主机, 提供了一种基于网络的存储解决方案。

### 1.3.4 iSCSI 目标门户

iSCSI 目标门户即 HBlock 服务器的目标门户, 用于与 HBlock 不在同一局域网的 Initiator 通信。

如果与 HBlock 不在同一局域网的 Initiator 想访问 HBlock 某一服务器, 需要先进行网络配置 (如 NAT 等), 确保 Initiator 可以通过该 IP 地址访问 HBlock 服务器, 然后将该地址配置为 HBlock 服务器的目标门户, 之后 Initiator 即可通过配置的目标门户访问 HBlock 服务器。

### 1.3.5 故障域

HBlock 将副本数据的各个副本或者 EC 模式数据的各个分块，按照故障域分配并存储，以达到数据保护的目的。同一数据的各个副本，以及同一数据的 EC 分块，写入不同故障域中。

### 1.3.6 数据服务

一个数据目录对应一个数据服务，HBlock 通过数据服务管理数据目录内用户的文件数据块。

### 1.3.7 监控

监控是指对 HBlock 系统、服务器、数据目录、卷的性能指标进行监测记录，用户可以查看实时或者历史性能数据，关注存储服务的性能。监控指标详见附录**监控指标**。

### 1.3.8 事件

事件指系统记录的用户操作 HBlock 的行为或 HBlock 系统行为，方便排查故障、审计和跟踪，全方面掌控存储运行情况。

事件分为用户事件和系统事件：

- 用户事件：用户操作 HBlock 的行为，具体用户事件列表详见附录**用户事件列表**。
- 系统事件：HBlock 系统行为，具体系统事件列表详见附录**系统事件列表**。

### 1.3.9 日志采集

日志采集是指用户主动采集 HBlock 日志数据，生成日志文件后下载到本地查看日志详细内容，以便排查故障。可以通过指定时间段、日志类型和服务器，缩小日志采集范围加快采集进度。

### 1.3.10 告警

告警指系统检测到 HBlock 业务或系统异常时产生的信息。

告警分为三个级别：

- 警告（Warning）：指一般性的，系统检测到潜在的或即将发生的影响业务的故障，当前还没有影响业务的告警。维护人员需及时查找告警原因，消除故障隐患。
- 重要（Major）：指局部范围内的、对系统性能造成影响的告警。需要尽快处理，否则会影响重要功能运行。
- 严重（Critical）：指带有全局性的、已经造成业务中断，或者会导致瘫痪的告警。需立即处理，否则系统有崩溃危险。

告警列表详见附录**告警列表**。

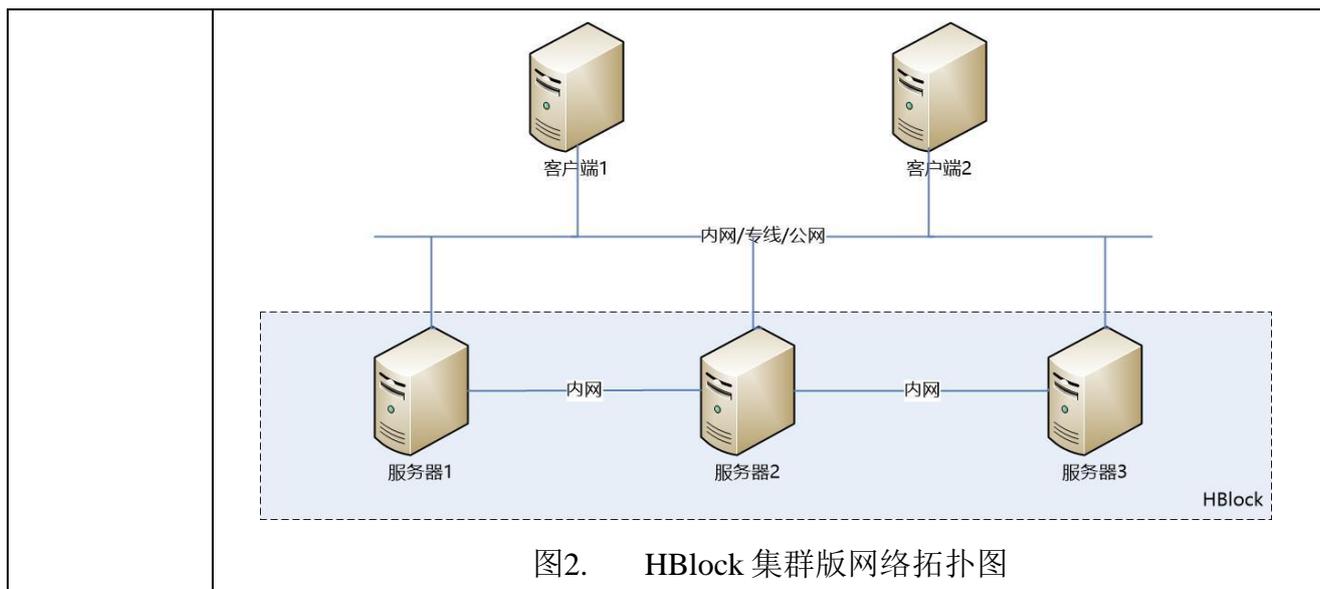
## 1.4 术语与缩略语

术语与缩略语	描述
ALUA	Asymmetric Logical Unit Access, 非对称逻辑单元访问。
CHAP	Challenge Handshake Authentication Protocol, 质询握手认证协议。
CIDR	Classless Inter-Domain Routing, 无类别域间路由, 一个用于给用户分配 IP 地址以及在互联网上有效地路由 IP 数据包的对 IP 地址进行归类的方法。
DSM	Device Specific Module, 设备特定模块。
EC	Erasur Coding, 纠删码。
IQN	iSCSI Qualified Name, iSCSI 限定名。
iSCSI	Internet Small Computer System Interface, 互联网小型计算机系统接口。
iSCSI initiator	iSCSI 发起程序。
I/O	Input/Output, 输入/输出。
LUN	Logical Unit Number, 逻辑单元号。
MPIO	Multipath I/O, 多路径 IO 管理。
NFS	Network File System, 网络文件系统。
NTP	Network Time Protocol, 网络时间协议。
RAID	Redundant Arrays of Independent Disks, 独立磁盘冗余阵列。
SAN	Storage Area Network, 存储区域网络。
SPC	SCSI (Small Computer System Interface, 小型计算机系统接口) Primary Commands, SCSI 基础命令。
SSD	Solid State Disk, 固态硬盘。
SSL	Secure Sockets Layer, 安全套接字协议。
Target	存储目标。
数据目录	用于存储 HBlock 数据的目录。

## 2 服务器部署要求

### 2.1 环境要求

项目	描述
支持 Linux OS	CentOS 7、8、9，CTyunOS 3。64 位操作系统。 ARM 架构的硬件环境下，推荐使用 PageSize 为 4k 的操作系统。
硬件	x86 服务器、ARM 服务器。 最低配置：单核 CPU、2GB 内存。可根据实际业务需要增加配置。
带宽	<ul style="list-style-type: none"> <li>● 客户端到 HBlock 的带宽：读写带宽能力大于业务读写带宽。</li> <li>● 数据目录对应磁盘分区的写带宽能力大于用户实际写入数据的带宽。</li> </ul>
安装目录所在盘	10GB 以上，建议配置为 RAID 1 或者 RAID 10。
数据目录	<ul style="list-style-type: none"> <li>● 最小配置：5GB，可根据实际业务需要增加配置。</li> <li>● 根据存储容量和副本模式配置数据目录对应分区的容量。</li> </ul> 对于 HBlock 使用到的目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。
网络设定	网络整体架构如下： <ol style="list-style-type: none"> <li>1. HBlock 内部各节点之间通过内网互联。</li> <li>2. HBlock 与上层应用之间通过内网或专线或公网互联。</li> </ol> <div style="text-align: center;">  <p>客户端                      HBlock服务器</p> </div> <p>图1. 单机版网络拓扑图</p>



**注意：**在部署 HBlock 前，需要明确使用**单机版**还是**集群版**，因为一旦部署后，不支持通过增减服务器进行模式切换。

## 2.2 HBlock 配置环境 – 单机版

按照环境要求，准备 1 台服务器。

**注意：** 确保 ping 命令和 ps 命令可用。Debian/Ubuntu 可以使用下列命令安装 ping 命令和 ps 命令。

```
apt-get update /*获取最新安装包
apt-get install iputils-ping /* 安装 ping 命令
apt-get install procps /* 安装 ps 命令安装
```

服务器按照下列操作步骤完成配置，以下操作以 CentOS 7.x 版本为例：

**说明：** 如果已经安装操作系统，请忽略步骤一。如果磁盘已挂载，请忽略步骤二，可以使用挂载路径作为 HBlock 的数据目录，或者使用命令 `mkdir DIRECTORY` 在挂载路径下创建一个目录，将此目录作为 HBlock 数据目录。

(一) 安装操作系统 CentOS 7.x 版本（可选）

(二) 格式化硬盘并挂载（可选）

请参考下列示例将您服务器上的硬盘进行格式化，方便后续部署使用。

```
lsblk #查看硬盘
mkfs.ext4 /dev/vdX #将硬盘格式化为 ext4
mkdir DIRECTORY #创建挂载路径，DIRECTORY 为路径名
mount /dev/vdX DIRECTORY #挂载硬盘
```

**说明：** mount 命令为临时挂载命令，服务器重启后，需要再次挂载。对于 HBlock 使用到的目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

(三) 防火墙设定

若您的服务器未开启防火墙，可以忽略此步骤。若您的服务器已开启防火墙，请开启 iSCSI 端口，以便客户端连接到服务器的 Target。示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

```
firewall-cmd --permanent --add-port=3260/tcp
```

2. 重新加载防火墙使配置生效：

```
firewall-cmd --reload
```

#### (四)设置资源限制

修改配置文件/etc/security/limits.conf，在配置文件中增加下列内容，设置在 *domain* 中打开的最大文件数。

```
domain soft nofile 65536 # 参数 domain 根据情况设置为具体的值
domain hard nofile 65536 # 参数 domain 根据情况设置为具体的值
```

可以根据情况，将 *domain* 设置为对应的 *username*、*groupname*、*uid*、*wildcard*。

**注意：**如果 *domain* 设置为对应的 *username*，则必须包含启动 HBlock 服务的用户。

- 例 1：例如 *domain* 取值为\*，表示所有用户打开的最大文件数为 65536。

```
* soft nofile 65536 # *为参数 domain 的取值
* hard nofile 65536 # *为参数 domain 的取值
```

- 例 2：例如 *domain* 取值为 root，表示 root 用户打开的最大文件数为 65536。

```
root soft nofile 65536 # root 为参数 domain 的取值
root hard nofile 65536 # root 为参数 domain 的取值
```

## 2.3 HBlock 配置环境 – 集群版

按照环境要求准备 3 台或 3 台以上的服务器。

**注意：**确保 ping 命令和 ps 命令可用。Debian/Ubuntu 可以使用下列命令安装 ping 命令和 ps 命令。

```
apt-get update /*获取最新安装包
apt-get install iputils-ping /* 安装 ping 命令
apt-get install procps /* 安装 ps 命令安装
```

每台服务器按照下列操作步骤完成配置，以下操作以 CentOS 7.x 版本为例：

**说明：**如果已经安装操作系统，请忽略步骤一。如果磁盘已挂载，请忽略步骤二，可以使用挂载路径作为 HBlock 的数据目录，或者使用命令 `mkdir DIRECTORY` 在挂载路径下创建一个目录，将此目录作为 HBlock 数据目录。

(一)安装操作系统 CentOS 7.x 版本（可选）

(二)格式化硬盘并挂载（可选）

请参考下列示例将服务器上的硬盘进行格式化，方便后续部署使用。

```
lsblk #查看硬盘
mkfs.ext4 /dev/vdX #将硬盘格式化为 ext4
mkdir DIRECTORY #创建挂载路径，DIRECTORY 为路径名
mount /dev/vdX DIRECTORY #挂载硬盘
```

**说明：**mount 命令为临时挂载命令，服务器重启后，需要再次挂载。对于 HBlock 使用到的目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

**注意：**如果安装 HBlock 的用户为非 root 用户，需要对 HBlock 使用到的目录有读写权限，可以使用下列命令。

```
chown HBlock 用户:HBlock 用户所属组 DIRECTORY
```

(三)防火墙设定

确保集群服务器之间可以相互访问，集群服务器之间相互添加白名单，另外请开启 iSCSI 端口，以便客户端连接到服务器的 Target。如果是在云主机上安装，安全组中也需要添加白名单。

若您的服务器未开启防火墙，可以忽略此步骤。

示例如下：

1. 开启 iSCSI 端口，如 iSCSI 端口为 3260 时：

```
firewall-cmd --permanent --add-port=3260/tcp
```

2. 集群各服务器的 IP 添加白名单：

- 添加 IPv4 地址

```
firewall-cmd --permanent --add-rich-rule="rule family=ipv4 source address=your_IP accept" // your_IP is IP address allowed to access
```

- 添加 IPv6 地址

```
firewall-cmd --permanent --add-rich-rule="rule family=ipv6 source address=your_IP accept" // your_IP is IP address allowed to access
```

3. 重新加载防火墙使配置生效：

```
firewall-cmd --reload
```

#### (四)设置资源限制

修改配置文件 `/etc/security/limits.conf`，在配置文件中增加下列内容，设置在 `domain` 中打开的最大文件数。

```
domain soft nofile 65536 # 参数 domain 根据情况设置为具体的值
domain hard nofile 65536 # 参数 domain 根据情况设置为具体的值
```

可以根据情况，将 `domain` 设置为对应的 `username`、`groupname`、`uid`、`wildcard`。

**注意：**如果 `domain` 设置为对应的 `username`，则必须包含启动 HBlock 服务的用户。

- 例 1：例如 `domain` 取值为 `*`，表示所有用户打开的最大文件数为 65536。

```
* soft nofile 65536 # *为参数 domain 的取值
* hard nofile 65536 # *为参数 domain 的取值
```

- 例 2: 例如 *domain* 取值为 root, 表示 root 用户打开的最大文件数为 65536。

```
root soft nofile 65536 # root 为参数 domain 的取值  
root hard nofile 65536 # root 为参数 domain 的取值
```

## 3 安装访问

### 3.1 安装

1. 请先完成以下准备工作：在服务器上准备一个或多个目录作为 HBlock 数据目录，用来存储 HBlock 数据。如：/mnt/storage01。对于集群版，每台服务器准备的目录可以不同。建议数据目录不要与操作系统共用磁盘或文件系统。
2. 将安装包放到服务器欲安装 HBlock 的目录下并解压缩，进入解压缩后的文件夹。

**说明：**建议安装目录不要与数据目录共用磁盘或文件系统。

```
unzip CTYUN_HBlock_Plus_3.5.0_x64.zip
cd 解压缩文件
```

**示例：**在服务器上执行解压缩安装包，并进入解压缩后的文件夹。

```
[root@hblockserver opt]# unzip CTYUN_HBlock_Plus_3.5.0_x64.zip
.....
[root@hblockserver opt]# cd CTYUN_HBlock_Plus_3.5.0_x64/
```

### 3. 安装 HBlock

在每台服务器上安装 HBlock。

```
./stor install [ { -a | --api-port } API_PORT ] [ { -w | --web-port } WEB_PORT ]
```

**API\_PORT：**指定 API 端口号，默认端口号为 1443。

**WEB\_PORT：**指定 WEB 端口号，默认端口号为 443。

**说明：**您可以根据业务需要设置 API 端口号和 WEB 端口号。

**注意：**请确保 Linux 用户具有所需要端口的权限。Linux 系统默认小于 1024 的端口不对没有 root 权限的 Linux 普通用户开放。

**示例：**在服务器上安装 HBlock。

```
[root@hblockserver1 CTYUN_HBlock_Plus_3.5.0_x64]# ./stor install
Do you agree with HBlock User Agreement
(https://www.ctyun.cn/portal/protocol/10073150)? [Yes/No]:
y
Installing HBlock...
Installed successfully.
When all servers are installed, please initialize HBlock in any of the following ways:
1. Use web portal to initialize HBlock. The https port is 443.
2. Use management API (POST /rest/v1/system/setup) to initialize HBlock. The https port
is 1443.
3. Use command line (stor setup) to initialize HBlock. Type 'stor --help setup' for more
information.
```

4. 安装完成：安装完成后，可以使用 WEB、命令行或者 API 对 HBlock 进行初始化及管理操作。本文后续将介绍使用 API 对 HBlock 进行初始化及管理操作。

## 3.2 访问地址

管理 API 服务启动后，可以通过 `https://SERVER_IP:PORT` 地址进行访问。仅支持 HTTPS 协议，`SERVER_IP` 可以是已安装 HBlock 的任意一台服务器 IP 地址（IPv4 或者 IPv6）。

参数	描述
<code>SERVER_IP</code>	安装 HBlock 的服务器 IP。 确保该服务器能够被您访问到，只有 API 客户端和被访问服务器在同一局域网内， <code>SERVER_IP</code> 才能使用服务器的内网 IP。
<code>PORT</code>	安装时配置的 API 端口，如果未配置，默认为 1443。

## 3.3 接口

### 3.3.1 URL 规则

采用 HTTP Restful API，URL 规则为：

```
https://{ip}:{port}/rest/{version}/{prefix}/{resourceName}/{name}[?({parameterName}={value})+]
```

#### 参数描述

参数	描述	是否必须
ip	服务器 IP。	是
port	管理 API 的端口。	是
version	版本号，目前版本号只能为 v1。	是
prefix	前缀，取值为： <ul style="list-style-type: none"> <li>● system: 系统相关接口。</li> <li>● block: 块操作相关接口。</li> </ul>	是
resourceName	资源名称。	是
name	具体实例的名称或 ID。	否

### 3.3.2 通用请求头

请求头	描述	是否必须
Host	请求的 IP 和端口。	是
Connection	是否需要持久连接，取值： <ul style="list-style-type: none"> <li>● keep-alive: 持续连接。</li> <li>● close: 关闭连接。</li> </ul> 默认值为keep-alive。	否
User-Agent	请求的客户端信息。	否
Date	请求发送的日期和时间，格式： <i>EEE, dd MMM yyyy HH:mm:ss 'GMT'</i> 。	是
Content-Type	请求体MIME类型，取值为： <code>application/json; charset=utf-8</code> 。	否
Content-Length	请求体的长度。	是
Content-MD5	按照RFC 1864，使用base64编码格式生成信息的128位MD5值。此请求头可以用作数据完整性检查，以验证数据是否与客户端发送的数据相同。	否
Authorization	包含请求的签名认证信息，格式是： <code>HB1ock+ " "+userName+": "+signature</code> 。	是

### 3.3.3 签名方法

Authorization 请求头中，计算签名的方法为：

Authorization: HBlock+ " "+*userName*+": "+*signature*, 其中：

- HBlock: 固定字符串。
- *userName*: HBlock 的管理员用户名。

**注意：**使用 API 初始化时，需要使用默认用户名 `storuser` 进行签名，初始化时可以修改用户名。

- *signature* = SHA256(Date 请求头的值+"\n"+SHA256(用户的 *password*))。

**注意：**使用 API 初始化时，需要使用初始密码进行签名，初始密码为 123456。初始化时必须重新设置密码，后续使用其他接口时使用新设置的密码进行签名。

### 3.3.4 通用响应头

响应头	描述
Server	服务名。
Connection	表示是否需要持久连接： <ul style="list-style-type: none"><li>● keep-alive: 持续连接。</li><li>● close: 关闭连接。</li></ul>
Date	响应时间。
Content-Type	响应体的MIME类型。
Content-Length	响应体的长度。
x-hblock-request-id	唯一标示请求的一个32位字符串，用于排查定位问题。
Retry-After	当服务端backoff时，返回此响应头，值为90。建议客户端90秒后重试。 <b>backoff 说明：</b> 当系统服务繁忙，暂不可用时，会返回客户端503响应状态码和Retry-After响应头。

### 3.3.5 通用错误码

HTTP status	错误码	错误信息	说明
400	BadDigest	The Content-MD5 you specified does not match what HBlock received.	Content-MD5 的值和请求体的 MD5 不一致。
400	ClusterModeNotAllowed	'operation' is not supported by cluster mode of HBlock.	集群版本 HBlock 不支持该操作。
400	IncompleteBody	You did not provide the number of bytes specified by the Content-Length HTTP header.	请求体长度不足 Content-Length 请求头指定的长度。
400	InsufficientServerSpace	This method is not allowed because the free space of <i>directory</i> must be greater than or equal to <i>value</i> for server <i>serverId[, serverId...]</i> .	服务器存储空间不足，无法执行操作。
400	InvalidArgument	Value empty at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be empty.	Json 数组长度不能为 0。
400	InvalidBoolean	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be boolean.	布尔类型不合法。
400	InvalidFilter	Value <i>value</i> at ' <i>filter</i> ' failed to satisfy constraint: Argument must use a single colon (':' means fuzzy match) or two colons ( '::' means	filter 参数不合法。

		exact match) to separate the attribute key and value. If you query multiple attributes, you can use 'and' or 'or' keywords to separate them, and there must be spaces before and after 'and' or 'or'.	
400	InvalidHeader	Invalid HTTP header, <i>headerName</i> .	请求头格式不合法。
400	InvalidInteger	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be integer.	值必须为整数。
400	InvalidRequest	Unable to parse request <i>url</i> .	URL 解析失败。
400	InvalidRequestBody	Invalid request body.	请求无效。
400	InvalidStatus	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [Enabled, Disabled].	输入的参数值不对。
400	InvalidString	Value " at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be an empty string.	参数不能是空字符串。
400	LoadConfigFailed	Failed to load configuration files.	加载配置文件失败。 加载配置文件需要满足以下条件： <ul style="list-style-type: none"> <li>● 需要基础服务器至少有 2 台满足 cs 和 ls 服务成功</li> </ul>

			启动。 <ul style="list-style-type: none"> <li>● 需要基础服务器至少有 1 台满足 fc、mdm 和 ds 服务成功启动。</li> <li>● 需要保证待启动的服务器和主服务器（Master）网络通信正常。</li> </ul>
400	MissingArgument	Value null at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be null.	Json 参数不能为空。
400	MissingHeader	Missing HTTP header <i>headerName</i> .	请求头 Host 不能为空。
400	NotInitialized	HBlock is not initialized.	HBlock 未初始化。
400	NotInstalled	HBlock is not installed.	HBlock 未安装。
400	RequestTimeout	No response received. Please check if there are any network issues and if the operation was successfully executed.	未收到响应，请检查是否存在网络问题，以及操作是否执行成功。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版不支持该操作。
403	LicenseExpired	The software license has expired.	软件许可证过期。
403	PermissionDenied	Failed to <i>action</i> because user has no permission.	用户的权限不足导致操作失败。
403	SignatureDoesNotMatch	Signature does not match. Please check your username, password	签名不正确。请检查用户名、密码和签名

		and signing method for server(s) [, <i>serverIP</i> ...].	方法。
403	TrialExpired	The software trial version has expired.	软件试用期已过期。
405	InvalidMethod	The specified method does not exist.	指定的方法不存在。
409	InvalidLUNStatus	The LUN status of ' <i>lunname</i> ' is ' <i>status</i> ', the request is invalid.	LUN当前的状态不正确，请求无效。
409	InvalidServerStatus	The server status is ' <i>status</i> ', the request is invalid.	服务器当前的状态不正确，请求无效。
409	InvalidStorStatus	The HBlock status is ' <i>status</i> ', the request is invalid.	HBlock当前的状态不正确，请求无效。
409	InvalidTargetStatus	The Target status is ' <i>status</i> ', the request is invalid.	Target当前的状态不正确，请求无效。
500	InsufficientResource	Operation failed due to insufficient resources.	内存或其他资源不足，操作失败。
500	InternalError	HBlock encountered an internal error. Please try again or contact the software vendor.	请重试或者联系软件供应商。
503	SlowDown	Please reduce your request rate.	服务端繁忙，请降低请求频率。

## 4 管理 API

### 4.1 初始化

#### 4.1.1 初始化

此操作用来初始化 HBlock，如果服务端返回 202 响应码，则表示开始执行初始化操作。

说明：安装完所有服务器后，需要初始化 HBlock。

注意：

- 使用 API 初始化时，需要使用默认用户名和初始密码进行签名，默认用户名为 `storuser`，初始密码为 `123456`。初始化时必须重新设置密码，后续使用其他接口时使用新设置的密码进行签名。
- 请确保 Linux 用户具有所需要端口的权限。Linux 系统默认小于 1024 的端口不对没有 root 权限的 Linux 普通用户开放。
- 设置端口范围（`portRange`）时，请避免和 Linux 系统的本地临时端口（`ip_local_port_range`）范围重合，否则可能会导致 HBlock 服务所用的端口被占用。使用命令行 `cat /proc/sys/net/ipv4/ip_local_port_range` 可以查看本地临时端口范围。

- 请求语法

单机版：

```
POST /rest/v1/system/setup HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "storName": storName,
  "userName": userName,
```

```
"newPassword": newPassword,
"iSCSIPort": iSCSIPort,
"portRange": port1-port2,
"ports":{
    "managementPort1": managementPort1,
    "managementPort2": managementPort2,
    "managementPort3": managementPort3,
    "managementPort4": managementPort4,
    "managementPort6": managementPort6
},
"servers":[
    {
        "ip": ip,
        "apiPort": apiPort,
        "diskPaths": [
            {"path": path,
             "capacityQuota": capacityvalue
            },
            {"path": path,
             "capacityQuota": capacityvalue
            },
            .....
        ]
    }
],
"publicNetwork": [publicNetwork]
}
```

### 集群版:

```
POST /rest/v1/system/setup HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization
```

```
{
  "storName": storName,
  "userName": userName,
  "newPassword": newPassword,
  "iSCSIPort": iSCSIPort,
  "portRange": port1-port2,
  "faultDomain": "faultDomain",
  "ports":{
    "dataPort1": dataPort1,
    "managementPort1": managementPort1,
    "managementPort2": managementPort2,
    "managementPort3": managementPort3,
    "managementPort4": managementPort4,
    "managementPort5": managementPort5,
    "managementPort6": managementPort6,
    "metadataPort1": metadataPort1,
    "metadataPort2": metadataPort2,
    "metadataPort3": metadataPort3,
    "metadataPort4": metadataPort4,
    "metadataPort5": metadataPort5,
    "metadataPort6": metadataPort6,
    "metadataPort7": metadataPort7,
    "metadataPort8": metadataPort8
  },
  "servers":[
    {
      "ip": ip,
      "apiPort": apiPort,
      "diskPaths": [
        {
          "path": path,
          "capacityQuota": capacityvalue
        },
        {
          "path": path,
          "capacityQuota": capacityvalue
        }
      ]
    }
  ]
}
```

```

        .....
    ]
},
.....
{
  "ip": ip,
  "apiPort": apiPort,
  "diskPaths": [
    {"path": path,
     " path ": capacityvalue
    },
    {"path": path,
     "capacityQuota": capacityvalue
    },
    .....
  ]
}
],
"services":{
  "mdm": [ip,ip],
  "ls": [ip,ip,ip],
  "cs": [ip,ip,ip]
},
"clusterNetwork": [clusterNetwork],
"publicNetwork": [publicNetwork]
}
    
```

● 请求参数

参数	描述	是否必须
storName	HBlock 名称。 类型：字符串 取值：长度范围是 1~64，可以包含字母、数字、下划线（_）和短横线（-），字母区分大小写，且仅	是

	支持以字母或数字开头。	
userName	<p>HBlock 的管理员用户名。</p> <p>类型：字符串</p> <p>取值：长度范围是 5~16，只能由数字和字母组成，字母区分大小。默认值为 storuser。</p>	否
newPassword	<p>设置的新密码。初始化时必须修改密码。</p> <p>类型：字符串</p> <p>取值：长度范围是 8~16，可包含字母、数字、特殊字符（~!@#\$%^&amp;*()_+[]{ };:.,/&lt;&gt;?），字母区分大小写。</p>	是
iSCSIPort	<p>iSCSI 端口。</p> <p>类型：整型</p> <p>取值：[1, 65535]，默认值为 3260。</p>	否
portRange	<p>指定端口范围。存储服务以及未指定端口的服务将从此范围中自动取值。</p> <p>类型：整型</p> <p>取值：[1, 65535]，<i>port1</i> 为端口范围最小值，<i>port2</i> 为端口范围最大值，且 <i>port1</i>&lt;<i>port2</i>。<i>port1</i> 默认取值为 20000，<i>port2</i> 默认取值为 20500。</p> <p><b>说明：</b>建议指定的端口范围至少包含 500 个端口。</p>	否
faultDomain	<p>指定集群故障域级别（仅集群版支持）。系统初始化之后，故障域不可修改。</p> <p>类型：枚举。</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>path</b>：数据目录级别的故障域。</li> <li>● <b>server</b>：服务器级别的故障域。</li> </ul>	否

	默认值为 <code>server</code> 。	
<code>ports.dataPort1</code>	数据端口 1（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
<code>ports.managementPort1</code>	管理服务端口 1。 类型：整型 取值：[1, 65535]。	否
<code>ports.managementPort2</code>	管理服务端口 2。 类型：整型 取值：[1, 65535]。	否
<code>ports.managementPort3</code>	管理服务端口 3。 类型：整型 取值：[1, 65535]。	否
<code>ports.managementPort4</code>	管理服务端口 4。 类型：整型 取值：[1, 65535]。	否
<code>ports.managementPort5</code>	管理服务端口 5（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
<code>ports.managementPort6</code>	管理服务端口 6。 类型：整型 取值：[1, 65535]。	否
<code>ports.metadataPort1</code>	元数据端口 1（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
<code>ports.metadataPort2</code>	元数据端口 2（仅集群版支持）。 类型：整型	否

	取值：[1, 65535]。	
ports.metadataPort3	元数据端口 3（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort4	元数据端口 4（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort5	元数据端口 5（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort6	元数据端口 6（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort7	元数据端口 7（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
ports.metadataPort8	元数据端口 8（仅集群版支持）。 类型：整型 取值：[1, 65535]。	否
servers	HBlock 服务器的属性集合，包含：ip、apiPort、diskPaths。 类型：数组	是
ip	HBlock 的服务器 IP。 取值：IPv4 或 IPv6 地址。	是
apiPort	管理 API 端口。 类型：整型 取值：[1, 65535]，默认值为 1443。需要和该服务器	否

	安装 HBlock 时设置的 API 端口号保持一致。	
diskPaths	数据目录属性集合。包括 path、 capacityQuota。 类型：数组。	是
path	指定数据目录，支持多个数据目录，至少指定一个数据目录。数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。 <b>说明：</b> 对于单机版，第一个数据目录为默认数据目录。 <b>注意：</b> 数据目录中不能有逗号（,）。 类型：字符串列表	是
capacityQuota	指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 类型：长整型。 取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。 <b>注意：</b> 如果相同的数据目录出现多次，以第一次出现的数据目录的容量配额为准。	否
services	服务地址的集合（仅集群版支持）。包括 mdm、ls、cs。 类型：数组。	否
mdm	元数据管理服务的地址（仅集群版支持）。如果填写，必须填写集群中的 2 个 IP 地址。如果不填写，元数据管理服务的地址默认为集群中的前 2 个 IP 地	否

	址。 取值：IPv4 或 IPv6 地址。	
ls	日志服务的地址（仅集群版支持）。如果填写，必须填写集群中的 3 个 IP 地址。日志服务的地址默认为集群中的前 3 个 IP 地址。 取值：IPv4 或 IPv6 地址。	否
cs	协调服务的地址（仅集群版支持）。如果填写，必须填写集群中的 3 个 IP 地址。协调服务的地址默认为集群中的前 3 个 IP 地址。 取值：IPv4 或 IPv6 地址。	否
clusterNetwork	集群网络（仅集群版支持），用于集群间的数据通信。 <ul style="list-style-type: none"> <li>● 如果指定集群网络，为了保证 HBlock 的各个服务器之间能够正常通信，请确保每个服务器上都有与指定网段相符的 IP，系统会自动选取该 IP 进行通信。</li> <li>● 如果未指定集群网络，默认使用服务器列表中 HBlock 服务器 IP，此时 HBlock 的服务器 IP 不能指定为 localhost、127.0.0.1 或 0:0:0:0:0:0:1。</li> </ul> 类型：CIDR 格式 取值：如果指定地址范围，IP 地址后加掩码表示，如 192.163.1.5/24。	否
publicNetwork	业务网络，用于客户端和 HBlock 之间的数据传输式。 <ul style="list-style-type: none"> <li>● 如果指定业务网络，请确保每个服务器上都有与指定网段相符的 IP，系统会自动选取该 IP 和</li> </ul>	否

	<p>客户端进行通信。</p> <ul style="list-style-type: none"> <li>● 如果未指定业务网络，或指定了业务网络，但指定的网段与服务器所有 IP 均不相符，默认使用服务器列表中 HBlock 服务器 IP，此时 HBlock 服务器 IP 不能指定为 localhost、127.0.0.1 或 0:0:0:0:0:0:1。</li> </ul> <p>类型：CIDR 格式</p> <p>取值：如果指定地址范围，IP 地址后加掩码表示，如 192.163.1.5/24 列表。</p>	
--	--	--

- 请求示例 1

单机版：初始化。

```

POST /rest/v1/system/setup HTTP/1.1
Date: Mon, 11 Dec 2023 06:06:47 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 543
Host: 192.168.0.32:1443
Authorization: HBlock storuser:signature

{
  "storName": "stor1",
  "newPassword": "*****",
  "portRange": "20000-20500",
  "iSCSIPort": 3260,
  "servers": [
    {
      "ip": "192.168.0.32",
      "apiPort": 1443,
      "diskPaths": [
        {
          "path": "/mnt/storage01",
          "capacityQuota": 96636764160
        }
      ]
    }
  ]
}
    
```

```
    },  
    {  
      "path": "/mnt/dskb",  
      "capacityQuota": 96636764160  
    }  
  ]  
}  
]  
}
```

- 响应示例 1

```
HTTP/1.1 202 Accepted  
x-hblock-request-id: 6589a0abcbde4bc5a4db3a6b9315008f  
Connection: keep-alive  
Date: Mon, 11 Dec 2023 06:06:47 GMT  
Content-Length: 0  
Server: HBlock
```

- 请求示例 2

集群版：初始化。

```
POST /rest/v1/system/setup HTTP/1.1  
Date: Wed, 24 Jan 2024 06:59:48 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 1344  
Host: 192.168.0.110:1443  
Authorization: HBlock storuser:signature  
  
{  
  "storName": "stor1",  
  "userName": "storuser1",  
  "newPassword": "*****",  
  "iSCSIPort": 3260,
```

```
"portRange": "20000-20500",
"faultDomain": "path",
"servers": [
  {
    "ip": "192.168.0.110",
    "apiPort": 1443,
    "diskPaths": [
      {
        "path": "/mnt/storage01",
        "capacityQuota": 96636764160
      }
    ]
  },
  {
    "ip": "192.168.0.192",
    "apiPort": 1443,
    "diskPaths": [
      {
        "path": "/mnt/stor",
        "capacityQuota": 96636764160
      }
    ]
  },
  {
    "ip": "192.168.0.102",
    "apiPort": 1443,
    "diskPaths": [
      {
        "path": "/mnt/stor",
        "capacityQuota": 96636764160
      }
    ]
  }
],
"services": {
```

```
    "mdm": [
      "192.168.0.110",
      "192.168.0.192"
    ],
    "ls": [
      "192.168.0.110",
      "192.168.0.192",
      "192.168.0.102"
    ],
    "cs": [
      "192.168.0.110",
      "192.168.0.192",
      "192.168.0.102"
    ]
  }
}
```

- 响应示例 2

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 096e457a5c26465590ff8f312a493f2c
Connection: keep-alive
Date: Wed, 24 Jan 2024 06:59:48 GMT
Content-Length: 0
Server: HBlock
```

## 4.1.2 查询初始化进度

此操作用来查询初始化进度。

### ● 请求语法

```
GET /rest/v1/system/setup HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

说明：本 API 请求不校验签名。

### ● 响应结果

名称	描述
progress	初始化进度百分比，范围为 0~100。初始化成功或者失败，进度都为 100。 可以通过返回的 status 来查看初始化是否成功。
status	初始化状态： <ul style="list-style-type: none"> <li>● Uninitialized: 未初始化。</li> <li>● Processing: 初始化中。</li> <li>● Succeeded: 初始化成功。</li> <li>● Failed: 初始化失败。</li> </ul>

### ● 请求示例

查询初始化进度。

```
GET /rest/v1/system/setup HTTP/1.1
Date: Mon, 14 Mar 2022 10:21:44 GMT
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature
```

### ● 响应示例

```
HTTP/1.1 200 OK
Date: Mon, 14 Mar 2022 10:21:44 GMT
Content-Type: application/json;charset=utf-8
Content-Length: 46
Connection: keep-alive
x-hblock-request-id: fa25dee228334c70bd05c2edf46393a6
Server: HBlock

{
  "data": {
    "progress": 25,
    "status": "Processing"
  }
}
```

### 4.1.3 错误码列表

HTTP status	错误码	错误信息	描述
400	BadUserName	Value ' <i>value</i> ' at ' <i>argument</i> ' failed to satisfy constraint: Argument can only contain letters and digits, and has length between 5 and 16.	用户名不合法。取值：字符串形式，长度范围是 5~16，只能由数字和字母组成，字母区分大小。
400	CanNotConnectToIP	Can not connect to the IP <i>IP</i> .	无法连接指定的 IP。
400	CanNotRemoveServer	The server can not be removed. Please stop the server, prepare a new server with same IP, and recover it.	无法移除集群中的基础节点服务器。
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接指定的服务器。
400	ClusterNetworkDoesNotMatch	All IPs on the server <i>serverIP</i> do not match the 'cluster network' <i>clusterNetwork</i> .	服务器的 IP 和 cluster network 不匹配。
400	DuplicateIP	The IP <i>IP</i> of ' <i>argument</i> ' is duplicated.	服务的 IP 地址重复。
400	InconsistentIPformat	The server IPs in the cluster should be unified as IPv4 or IPv6 format when cluster network is not specified.	如果不指定集群网络，集群内服务器 IP 应统一为 IPv4 或 IPv6 格式。

400	InsufficientPorts	Insufficient available ports, need to increase the port range by $N$ ports.	端口范围中可用端口数量不足，请调整范围使之至少增加 $N$ 个端口。
400	InvalidCIDR	Value <i>CIDR</i> at ' <i>networkType</i> network' failed to satisfy constraint: Argument does not match the IP of the ' <i>localhost</i> '/'127.0.0.1'/'0:0:0:0:0:0:0:1' server.	当 server ip 包含 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1 时，用户指定的 ' <i>cluster/public</i> network' 和服务器的 IP 不匹配。
400	InvalidClusterNetwork	Value <i>clusterNetwork</i> at ' <i>cluster</i> network' failed to satisfy constraint: Argument must satisfy CIDR specifications.	集群网格式错误。
400	InvalidDiskPath	The disk paths do not exist: <i>serverIP:diskPaths</i> [ <i>,diskPaths...</i> ] [ <i>;</i> <i>serverIP:diskPaths[,diskPaths...</i> ]]. The disk paths must be greater than or equal to 1 GiB: <i>serverIP:diskPaths</i> [ <i>,diskPaths...</i> ] [ <i>;</i> <i>serverIP:diskPaths[,diskPaths...</i> ]].	初始化过程中，因数据目录有问题而导致的失败。

		<p>The disk paths can not access:  <i>serverIP:diskPaths</i>  <i>[,diskPaths...] [;</i>  <i>serverIP:diskPaths[,diskPaths...</i>  <i>]].</i></p> <p>The disk paths must be                  directory: <i>serverIP:diskPaths</i>  <i>[,diskPaths...] [;</i>  <i>serverIP:diskPaths[,diskPaths...</i>  <i>]].</i></p> <p>The disk paths include                  incompatible data:  <i>serverIP:diskPaths</i>  <i>[,diskPaths...] [;</i>  <i>serverIP:diskPaths[,diskPaths...</i>  <i>]].</i></p> <p>The disk paths can not contain                  commas:  <i>serverIP:diskPaths</i>  <i>[,diskPaths...] [;</i>  <i>serverIP:diskPaths[,diskPaths...]</i>  <i>].</i></p>	
400	InvalidDiskPathCapacityQuota	<p>The capacity quota for the                  following disk paths failed to                  satisfy constraint. It must be                  integer and not greater than the                  total capacity of disk path.</p>	<p>容量配额参数错误。</p>

		<code>serverIP:diskPaths</code> <code>[,diskPaths...]</code> <code>[;serverIP:diskPaths[,diskPaths..</code> <code>.].</code> <code>].</code>	
400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [ <i>value1</i> , <i>value2</i> ... ]	枚举参数不合法。
400	InvalidIP	Value <i>IP</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy IPv4 or IPv6 specifications.	IP 格式错误。
400	InvalidLong	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be of type long.	参数填写错误，必须为整型。
400	InvalidNewPassword	Value ' <i>value</i> ' at 'new password' failed to satisfy constraint: Argument can contain letters, digits, special symbols (~! @ # \$% ^ & * () _ + [] {}   ; : , / < > ?) and has length between 8 and 16.	新密码不符合规则。
400	InvalidPort	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must have value between 1 and 65535.	Port 的取值必须在 [1, 65535] 之间。

400	InvalidPortRange	Value value at 'port range' failed to satisfy constraint: Argument must satisfy pattern 'port1-port2', where port1, port2 are positive integers between 1 and 65535, port1 is less than port2.	参数必须是“port1-port2”格式，port1 和 port2 必须是介于 [1,65535] 之间的正整数，且 port1 小于 port2。
400	InvalidPublicNetwork	Value <i>publicNetwork</i> at 'public network' failed to satisfy constraint: Argument must satisfy CIDR specifications.	业务网格式错误。
400	InvalidServers	The current server is not in the servers list.	当前服务器不在 servers 列表内。
400	InvalidServersCount	The number of servers for <i>xx</i> service should be <i>value</i> .	XX 服务的服务器数量应该为 <i>value</i> 台。
400	InvalidStorName	Value ' <i>value</i> ' at ' <i>argument</i> ' failed to satisfy constraint: Argument can only contain letters, digits, hyphens (-), and underscores(_), begin with a letter or digit, and does not exceed 64 characters.	HBlock 名称不合法。
403	InvalidUserName	Invalid user name.	用户名不正确。
403	InvalidUserNameOrPassword	Invalid user name or password hash.	用户名或密码不正确。
400	MissingClusterNetwork	Value null at 'cluster network' failed to satisfy constraint:	当 server ip 包含 localhost、

		Argument must not be null when 'server ip' contains 'localhost'/'127.0.0.1'/'0:0:0:0:0:0:0:1'.	127.0.0.1 或 0:0:0:0:0:0:0:1 时，集群网不能为空。
400	MissingPublicNetwork	Value null at 'public network' failed to satisfy constraint: Argument must not be null when 'server ip' contains 'localhost'/'127.0.0.1'/'0:0:0:0:0:0:0:1'.	当 server ip 包含 localhost、127.0.0.1 或 0:0:0:0:0:0:0:1 时，业务网不能为空。
400	ProductTypeDoesNotMatch	HBlock product type does not match.	产品类型不一致。
400	ServerAlreadyInitialized	The server has been initialized.	服务器已经初始化过
400	StandaloneModeNotAllowed	'operation' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	TooFewServers	The number of servers must be greater than or equal to 3.	服务器必须大于等于 3 台。
400	UnrecognizedServer	Value <i>IP</i> at 'argument' failed to satisfy constraint: Argument must be in the setup server list.	服务器 IP 不在初始化服务器列表内。
400	VersionDoesNotMatch	Please replace the HBlock version of server <i>serverIP</i> [, <i>serverIP</i> ...] to <i>clutserVersion</i> , then try again.	版本号不一致。
409	PortConflict	The following port is/ports are in use.	端口冲突。

		<i>server_ID/IP/local server: port_name port[, port_name port...]</i>  <i>[ server_ID/IP/local server: port_name port[, port_name port...]....]</i>	
--	--	---	--

## 4.2 软件许可证

### 4.2.1 导入软件许可证

此操作用来导入软件许可证。

#### ● 请求语法

```
POST /rest/v1/system/license HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "key": "key"
}
```

#### ● 请求参数

参数	描述	是否必须
key	软件许可证密钥。	是

#### ● 响应结果

名称	描述
licenseId	软件许可证 ID。
account	软件许可证所属的账号。
type	软件许可证的购买类型： <ul style="list-style-type: none"> <li>● Subscription: 订阅模式。</li> <li>● Perpetual: 永久许可模式。</li> </ul>
status	软件许可证的状态：

	<ul style="list-style-type: none"> <li>● <b>Effective:</b> 已生效。</li> <li>● <b>Expired:</b> 已过期。</li> <li>● <b>Invalid:</b> 已失效。</li> </ul> 如果软件许可证是 <b>Perpetual</b> 类型的，不存在过期状态。
<b>maximumLocalCapacity</b>	当前时间允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
<b>records</b>	软件许可证购买记录的集合，包含： <b>purchaseTime</b> 、 <b>operationtype</b> 、 <b>localCapacity</b> 、 <b>subscribeEffectiveTime</b> 、 <b>subscribeExpireTime</b> 、 <b>maintenanceEffectiveTime</b> 、 <b>maintenanceExpireTime</b> 、 <b>substatus</b> 。 类型：数组
<b>purchaseTime</b>	记录购买的时间，unix 时间戳（UTC），精确到毫秒。
<b>operationtype</b>	软件许可证的购买操作记录： <ul style="list-style-type: none"> <li>● <b>New:</b> 首次购买。</li> <li>● <b>Expand:</b> 扩容。</li> <li>● <b>Renew:</b> 续订/续保。</li> </ul>
<b>localCapacity</b>	本次购买对应的本地卷容量，单位 TiB。如果不限容量，不返回此项。
<b>subscribeEffectiveTime</b>	对于订阅模式的软件许可证，本次购买操作对应的软件许可证生效时间，unix 时间戳（UTC），精确到毫秒。
<b>subscribeExpireTime</b>	对于订阅模式的软件许可证，本次购买操作对应的软件许可证的过期时间，unix 时间戳（UTC），精确到毫秒。
<b>maintenanceEffectiveTime</b>	对于永久许可模式的软件许可证，本次购买操作对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
<b>maintenanceExpireTime</b>	对于永久许可模式的软件许可证，本次购买操作对应的维保的过期时间，unix 时间戳（UTC），精确到毫秒。

substatus	本次购买操作的当前生效状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>
usages	软件许可证的使用记录的集合，包含： maximumLocalCapacity、subscribeEffectiveTime、 subscribeExpireTime、maintenanceEffectiveTime、 maintenanceExpireTime、maintenanceExpireTime、 substatus。 类型：数组
maximumLocalCapacity	对应时间段内允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	对于订阅模式的软件许可证，许可证对应的生效时间，unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	对于订阅模式的软件许可证，许可证对应的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	对于永久许可模式的软件许可证，许可证对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	对于永久许可模式的软件许可证，许可证对应的维保过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	对于订阅模式的软件许可证，表示对应时间段内许可证的状态；对于永久许可模式的软件许可证，表示对应时间段内许可证的维保状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>

- 请求示例

导入软件许可证。

```
POST /rest/v1/system/license HTTP/1.1
Date: Thu, 17 Mar 2022 08:26:28 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 876
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature

{
  "key": "xxxxxxxxxxxxxxxx"
}
```

- 响应示例

```
HTTP/1.1 200 OK
Date: Thu, 17 Mar 2022 08:26:28 GMT
Content-Length: 658
Connection: keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: 24f2b6f45baf49a69fa9d0bfad1d76f3
Server: HBlock

{
  "data": {
    "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",
    "account": "xxx@testcom.cn",
    "status": "Effective",
    "type": "Subscription",
    "maximumLocalCapacity": 2048,
    "records": [
      {
        "purchaseTime": 1647505314201,
        "operationtype": "New",
        "localCapacity": 2048,

```

```
    "subscribeEffectiveTime": 1647505314201,  
    "subscribeExpireTime": 1655281314201,  
    "substatus": "Effective"  
  },  
  {  
    "purchaseTime": 1650097314201,  
    "operationtype": "Renew",  
    "localCapacity": 2048,  
    "subscribeEffectiveTime": 1655281314201,  
    "subscribeExpireTime": 1663057314201,  
    "substatus": "NotStart"  
  }  
],  
"usages": [  
  {  
    "maximumLocalCapacity": 2048,  
    "subscribeEffectiveTime": 1647505314201,  
    "subscribeExpireTime": 1663057314201,  
    "substatus": "Effective"  
  }  
]  
}
```

## 4.2.2 批量查询软件许可证信息

批量查询软件许可证信息，信息包括软件许可证的 record 信息以及 usage 信息。

record 信息：购买记录的信息。

usage 信息：软件许可证在不同时间段内的对应的容量和状态信息。

### ● 请求语法

```
GET /rest/v1/system/license?filter=filter&range=i-j&show=show HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization:authorization
```

### ● 请求参数

名称	描述	是否必须
filter	设置查询的过滤条件。支持的查询类型包括： <ul style="list-style-type: none"> <li>● licenseId: 软件许可证 ID。</li> <li>● status: 软件许可证的状态：                             <ul style="list-style-type: none"> <li>➢ Effective: 已生效。</li> <li>➢ Expired: 已过期。</li> <li>➢ Invalid: 无效。</li> </ul> </li> <li>● type: 软件许可证的购买类型：                             <ul style="list-style-type: none"> <li>➢ Subscription: 订阅模式。</li> <li>➢ Perpetual: 永久许可模式。</li> </ul> </li> </ul>	否
range	查询范围，格式：i-j，其中 i,j 是正整数，i 小于 j，j-i<1000。 按照 license 的购买时间由新到旧降序排列，返回第 i 个到第 j 个软件许可证。 如果输入的查询范围超过软件许可证的总个数，那么返回空结	否

	果集。 如果不输入 <code>range</code> 参数，那么返回所有软件许可证。	
<code>show</code>	查询的输出结果，支持的输出结果包括 <code>record</code> , <code>usage</code> 。 如果不填写，就是默认两个信息都输出。	否

### ● 响应结果

名称	描述
<code>licenseId</code>	软件许可证 ID。
<code>account</code>	软件许可证所属的账号。
<code>type</code>	软件许可证的购买类型： <ul style="list-style-type: none"> <li>● <code>Subscription</code>: 订阅模式。</li> <li>● <code>Perpetual</code>: 永久许可模式。</li> </ul>
<code>status</code>	软件许可证的状态： <ul style="list-style-type: none"> <li>● <code>Effective</code>: 已生效。</li> <li>● <code>Expired</code>: 已过期。</li> <li>● <code>Invalid</code>: 已失效。</li> </ul> 如果软件许可证是 <code>Perpetual</code> 类型的，不存在过期状态。
<code>maximumLocalCapacity</code>	当前时间允许的最大本地卷容量，单位 <code>TiB</code> 。如果不限容量，不返回此项。
<code>records</code>	软件许可证购买记录的集合，包含： <code>purchaseTime</code> 、 <code>operationtype</code> 、 <code>localCapacity</code> 、 <code>subscribeEffectiveTime</code> 、 <code>subscribeExpireTime</code> 、 <code>maintenanceEffectiveTime</code> 、 <code>maintenanceExpireTime</code> 、 <code>substatus</code> 。 类型：数组
<code>purchaseTime</code>	记录购买的时间，unix 时间戳（UTC），精确到毫秒。
<code>operationtype</code>	软件许可证的操作记录：

	<ul style="list-style-type: none"> <li>● <b>New:</b> 首次购买。</li> <li>● <b>Expand:</b> 扩容。</li> <li>● <b>Renew:</b> 续订/续保。</li> </ul>
<code>localCapacity</code>	本次购买对应的本地卷容量，单位 TiB。如果不限容量，不返回此项。
<code>subscribeEffectiveTime</code>	对于订阅模式的软件许可证，本次购买操作对应的软件许可证生效时间，unix 时间戳（UTC），精确到毫秒。
<code>subscribeExpireTime</code>	对于订阅模式的软件许可证，本次购买操作对应的软件许可证的过期时间，unix 时间戳（UTC），精确到毫秒。
<code>maintenanceEffectiveTime</code>	对于永久许可模式的软件许可证，本次购买操作对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
<code>maintenanceExpireTime</code>	对于永久许可模式的软件许可证，本次购买操作对应的维保的过期时间，unix 时间戳（UTC），精确到毫秒。
<code>substatus</code>	本次购买操作的当前生效状态： <ul style="list-style-type: none"> <li>● <b>NotStart:</b> 未生效。</li> <li>● <b>Effective:</b> 已生效。</li> <li>● <b>Expired:</b> 已过期。</li> </ul>
<code>usages</code>	软件许可证的使用记录的集合，包含： <code>maximumLocalCapacity</code> 、 <code>subscribeEffectiveTime</code> 、 <code>subscribeExpireTime</code> 、 <code>maintenanceEffectiveTime</code> 、 <code>maintenanceExpireTime</code> 、 <code>substatus</code> 。 类型：数组
<code>maximumLocalCapacity</code>	对应时间段内允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
<code>subscribeEffectiveTime</code>	对于订阅模式的软件许可证，许可证对应的生效时间，unix 时间戳（UTC），精确到毫秒
<code>subscribeExpireTime</code>	对于订阅模式的软件许可证，许可证对应的过期时间，unix

	时间戳（UTC），精确到毫秒。
<code>maintenanceEffectiveTime</code>	对于永久许可模式的软件许可证，许可证对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
<code>maintenanceExpireTime</code>	对于永久许可模式的软件许可证，许可证对应的维保过期时间，unix 时间戳（UTC），精确到毫秒。
<code>substatus</code>	对于订阅模式的软件许可证，表示对应时间段内许可证的状态；对于永久许可模式的软件许可证，表示对应时间段内许可证的维保状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>

### ● 请求示例

批量查询软件许可证信息。

```
GET /rest/v1/system/license?filter=type:Subscription HTTP/1.1
Date: Thu, 17 Mar 2022 09:40:42 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 453c0e4d909b4cb4801759b89f490bbc
Connection: keep-alive
Content-Length: 671
Date: Thu, 17 Mar 2022 09:40:42 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
```

```
"licenses": [  
  {  
    "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",  
    "account": "xxx@testcom.cn",  
    "status": "Invalid",  
    "type": "Subscription",  
    "maximumLocalCapacity": 2048,  
    "records": [  
      {  
        "purchaseTime": 1647505314201,  
        "operationtype": "New",  
        "localCapacity": 2048,  
        "subscribeEffectiveTime": 1647505314201,  
        "subscribeExpireTime": 1655281314201,  
        "substatus": "Effective"  
      },  
      {  
        "purchaseTime": 1650097314201,  
        "operationtype": "Renew",  
        "localCapacity": 2048,  
        "subscribeEffectiveTime": 1655281314201,  
        "subscribeExpireTime": 1663057314201,  
        "substatus": "NotStart"  
      }  
    ],  
    "usages": [  
      {  
        "maximumLocalCapacity": 2048,  
        "subscribeEffectiveTime": 1647505314201,  
        "subscribeExpireTime": 1663057314201,  
        "substatus": "Effective"  
      }  
    ]  
  }  
]
```

```
}  
}
```

### 4.2.3 查询指定软件许可证信息

查询指定软件许可证的信息。信息包括软件许可证的 record 信息以及 usage 信息：

- record 信息：购买记录的信息。
- usage 信息：软件许可证在不同时间段内的对应的容量和状态信息。

#### ● 请求语法

```
GET /rest/v1/system/license/LicenseId?show=show HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

名称	描述	是否必须
licenseId	软件许可证的 ID。	否
show	查询的输出结果，record、usage。 如果不填写，就是默认两个信息都输出。	否

#### ● 响应结果

名称	描述
licenseId	软件许可证 ID。
account	软件许可证所属的账号。
type	软件许可证的购买类型： <ul style="list-style-type: none"> <li>● Subscription: 订阅模式。</li> <li>● Perpetual: 永久许可模式。</li> </ul>
status	软件许可证的状态：

	<ul style="list-style-type: none"> <li>● <b>Effective:</b> 已生效。</li> <li>● <b>Expired:</b> 已过期。</li> <li>● <b>Invalid:</b> 已失效。</li> </ul> 如果软件许可证是 <b>Perpetual</b> 类型的，不存在过期状态。
<b>maximumLocalCapacity</b>	当前时间允许的最大本地卷容量，单位 <b>TiB</b> 。如果不限容量，不返回此项。
<b>records</b>	软件许可证的购买记录的集合，包含： <b>purchaseTime</b> 、 <b>operationtype</b> 、 <b>localCapacity</b> 、 <b>subscribeEffectiveTime</b> 、 <b>subscribeExpireTime</b> 、 <b>maintenanceEffectiveTime</b> 、 <b>maintenanceExpireTime</b> 、 <b>substatus</b> 。 类型：数组
<b>purchaseTime</b>	记录购买的时间， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>operationtype</b>	软件许可证的操作记录： <ul style="list-style-type: none"> <li>● <b>New:</b> 首次购买。</li> <li>● <b>Expand:</b> 扩容。</li> <li>● <b>Renew:</b> 续订/续保。</li> </ul>
<b>localCapacity</b>	本次购买对应的本地卷容量，单位 <b>TiB</b> 。如果不限容量，不返回此项。
<b>subscribeEffectiveTime</b>	对于订阅模式的软件许可证，本次购买操作对应的软件许可证生效时间， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>subscribeExpireTime</b>	对于订阅模式的软件许可证，本次购买操作对应的软件许可证的过期时间， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>maintenanceEffectiveTime</b>	对于永久许可模式的软件许可证，本次购买操作对应的维保生效时间， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。
<b>maintenanceExpireTime</b>	对于永久许可模式的软件许可证，本次购买操作对应的维保的过期时间， <b>unix</b> 时间戳（ <b>UTC</b> ），精确到毫秒。

substatus	本次购买操作的当前状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>
usages	软件许可证使用记录的集合，包含： maximumLocalCapacity、subscribeEffectiveTime、 subscribeExpireTime、 maintenanceEffectiveTime、 maintenanceExpireTime、substatus。 类型：数组
maximumLocalCapacity	对应时间段内允许的最大本地卷容量，单位 TiB。如果不限容量，不返回此项。
subscribeEffectiveTime	对于订阅模式的软件许可证，许可证对应的生效时间，unix 时间戳（UTC），精确到毫秒。
subscribeExpireTime	对于订阅模式的软件许可证，许可证对应的过期时间，unix 时间戳（UTC），精确到毫秒。
maintenanceEffectiveTime	对于永久许可模式的软件许可证，许可证对应的维保生效时间，unix 时间戳（UTC），精确到毫秒。
maintenanceExpireTime	对于永久许可模式的软件许可证，许可证对应的维保过期时间，unix 时间戳（UTC），精确到毫秒。
substatus	对于订阅模式的软件许可证，表示对应时间段内许可证的状态；对于永久许可模式的软件许可证，表示对应时间段内许可证的维保状态： <ul style="list-style-type: none"> <li>● <b>NotStart</b>: 未生效。</li> <li>● <b>Effective</b>: 已生效。</li> <li>● <b>Expired</b>: 已过期。</li> </ul>

- 请求示例

查询软件许可证 ID 为 ehc2b6a9-f3fb-4098-a6b3-3652a5d76269 的软件许可证信息。

```
GET /rest/v1/system/license/ehc2b6a9-f3fb-4098-a6b3-3652a5d76269 HTTP/1.1
Date: Thu, 17 Mar 2022 09:43:54 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: a63115d3bc704ef7a64a400c747a6b2d
Connection: keep-alive
Content-Length: 656
Date: Thu, 17 Mar 2022 09:43:54 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "licenseId": "ehc2b6a9-f3fb-4098-a6b3-3652a5d76269",
    "account": "xxx@testcom.cn",
    "status": "Invalid",
    "type": "Subscription",
    "maximumLocalCapacity": 2048,
    "records": [
      {
        "purchaseTime": 1647505314201,
        "operationtype": "New",
        "localCapacity": 2048,
        "subscribeEffectiveTime": 1647505314201,
        "subscribeExpireTime": 1655281314201,
        "substatus": "Effective"
      },
      {
```

```
    "purchaseTime": 1650097314201,  
    "operationtype": "Renew",  
    "localCapacity": 2048,  
    "subscribeEffectiveTime": 1655281314201,  
    "subscribeExpireTime": 1663057314201,  
    "substatus": "NotStart"  
  },  
],  
"usages": [  
  {  
    "maximumLocalCapacity": 2048,  
    "subscribeEffectiveTime": 1647505314201,  
    "subscribeExpireTime": 1663057314201,  
    "substatus": "Effective"  
  }  
]  
}  
}
```

#### 4.2.4 错误码列表

HTTP status	错误码	错误信息	描述
400	ExceedLocalCapacityLimit	The total capacity of local mode LUN can not exceed the maximum capacity limit agreed in the software license.	本地卷容量不能超过软件许可证规定的容量。
400	InvalidFilterEnum	Value <i>value</i> at 'filter' failed to satisfy constraint: Argument can only contain: [ <i>licenseId</i> , <i>status</i> , <i>type</i> ].	Filter 参数不正确。
400	InvalidLicenseKey	The license key is invalid.	软件许可证无效。
400	InvalidRange	Value <i>value</i> at 'range' failed to satisfy constraint: Argument must satisfy pattern 'i-j', where i, j are positive integers, i is less than j, and j-i<1000.	range 参数不正确。
400	InvalidShow	Value <i>value</i> at 'show' failed to satisfy constraint: Argument can only contain: [ <i>record</i> , <i>usage</i> ].	show 参数不正确。
400	LicenseKeyDoesNotMatch	This license key does not match the current HBlock. Please obtain the software	软件许可证密钥与当前 HBlock 不匹配。

		license again.	
400	ReimportLicenseKey	Do not reimport the license key that is in effect.	重复导入正在生效的软件许可证密钥。
400	SystemTimeTooEarly	System time is <i>time</i> , it cannot be less than the effective time <i>time</i> of the software license	系统时钟不能早于软件许可证的生效时间。
400	SystemTimeTooLate	System time is <i>time</i> , it cannot be later than the expiration time <i>time</i> of the software license	系统时钟不能晚于软件许可证的结束时间。
403	LicenseExpired	The software license has expired.	软件许可证过期。
403	LicenseMaintenanceExpired	The maintenance for software license has expired.	软件许可证过保。
404	NoSuchLicenseID	The software license with ID <i>ID</i> does not exist.	软件许可证不存在。

## 4.3 卷管理

### 4.3.1 创建卷

此操作用来创建卷。创建卷时，如果指定的 Target 名称不存在，则同时创建 Target。

说明：HBlock 上 Target IQN 的数量最多 32766 个。一个 Target 最多可以关联 256 个卷，但是一个卷只能被一个 Target 关联。

- 请求语法

#### 单机版

```
POST /rest/v1/block/lun HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "targetName": targetName,
  "capacity": capacity,
  "config": {
    "sectorSize": sectorSize,
    "writePolicy": writePolicy,
    "path": path
  }
}
```

#### 集群版

```
POST /rest/v1/block/lun HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
```

```

Host: ip:port
Authorization: authorization

{
  "lunName": LunName,
  "targetName": targetName,
  "capacity": capacity,
  "config":{
    "localStorageClass": LocalStorageClass,
    "minReplica": minReplica,
    "ECfragmentSize": ECfragmentSize,
    "sectorSize": sectorSize,
    "highAvailability": highAvailability,
    "writePolicy": writePolicy
  }
}
    
```

● 请求参数

参数	描述	是否必须
lunName	卷名称。 类型：字符串 取值：长度范围是 1~16，只能由字母、数字和短横线 (-) 组成，字母区分大小写，且仅支持以字母或数字开头。	是
targetName	Target 名称。 <b>说明：</b> 创建卷时，如果指定的 Target 名称不存在，那么同时创建 iSCSI Target。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是

<p>capacity</p>	<p>卷容量。</p> <p>类型：整型</p> <p>取值：[1,1048576]，单位是 GiB。</p>	<p>是</p>
<p>config.localStorageClass</p>	<p>卷冗余模式（仅集群版支持）。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● single-copy：单副本。</li> <li>● 2-copy：两副本。</li> <li>● 3-copy：三副本。</li> <li>● EC N+M：纠删码模式。其中 N、M 为正整数，<math>N &gt; M</math>，且 <math>N + M \leq 128</math>。表示将数据分割成 N 个片段，并生成 M 个校验数据。</li> </ul> <p>默认值为 EC 2+1。</p> <p>说明（以下场景均为集群可用的前提下）：</p> <ul style="list-style-type: none"> <li>● 创建 EC N+M 的卷时：                     <ul style="list-style-type: none"> <li>■ 服务器级别故障域：集群中可用节点台数不足 N+M 时，卷可以创建成功，但是卷数据将处于降级状态，允许故障的服务器台数不足 M，建议尽快添加服务器或修复故障服务器。可用故障域不足 N 时，可以创建卷，但无法写入数据，且系统会产生告警。</li> <li>■ 数据目录级别故障域：集群中可用数据目录数量不足 N+M 时，卷可以创建成功，但是卷数据将处于降级状态，允许故障的数据目录个数不足 M，建议尽快添加数据目录或修复故障数据目录。可</li> </ul> </li> </ul>	<p>否</p>

	<p>用故障域不足 N 时，可以创建卷，但无法写入数据，且系统会产生告警。</p> <ul style="list-style-type: none"> <li>● 创建副本模式的卷时：可用故障域数量大于等于 1，即可创建两副本、三副本的卷，卷数据将处于降级状态，建议尽快添加服务器或修复故障服务器。</li> </ul>	
<code>config.minReplica</code>	<p>最小副本数（仅集群版支持）。</p> <p>对于副本模式的卷，假设卷副本数为 X，最小副本数为 Y（Y 必须 <math>\leq X</math>），该卷每次写入时，至少 Y 份数据写入成功，才视为本次写入成功。</p> <p>对于 EC N+M 模式的卷，假设该卷最小副本数设置为 Y（必须满足 <math>N \leq Y \leq N+M</math>），必须满足总和至少为 Y 的数据块和校验块写入成功，才视为本次写入成功。</p> <p>类型：整型</p> <p>取值：对于副本卷，取值范围是 [1, N]，N 为副本模式卷的副本数，默认值为 1。对于 EC 卷，取值范围是 [N, N+M]，默认值为 N。</p>	否
<code>config.ECfragmentSize</code>	<p>纠删码模式分片大小（仅集群版支持）。</p> <p><code>config.localStorageClass</code> 为 EC 模式时，此设置才生效，否则忽略。</p> <p>取值：1、2、4、8、16、32、64、128、256、512、1024、2048、4096，单位是 KiB。默认值为 16。</p>	否
<code>config.sectorSize</code>	<p>扇区大小。</p> <p>类型：枚举</p> <p>取值：512、4096，单位为字节。默认值为</p>	否

	<p>4096。</p> <p><b>说明：</b>扇区大小的选取：根据自身业务场景，一般情况下，单次 I/O 操作的数据大小大于或接近 4KiB，则推荐选择 4096；单次 I/O 操作的数据大小接近 512Bytes，则推荐选择 512。</p>	
config.highAvailability	<p>是否开启高可用模式（仅集群版支持）。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>ActiveStandby：</b> 启用主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● <b>Disabled：</b> 不启用主备，该卷关联对应 Target 下的 1 个 IQN。</li> </ul> <p>默认值为 ActiveStandby。</p>	否
config.writePolicy	<p>卷的写策略。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>WriteBack：</b> 回写，指数据写入到内存后即返回客户端成功，之后再异步写入磁盘。适用于对性能要求较高，稳定性要求不高的场景。</li> <li>● <b>WriteThrough：</b> 透写，指数据同时写入内存和磁盘，并在都写成功后再返回客户端成功。适用于稳定性要求较高，写性能要求不高，且最近写入的数据会较快被读取的场景。</li> <li>● <b>WriteAround：</b> 绕写，指数据直接写到磁盘，不写入内存。适用于稳定性要求较高，</li> </ul>	否

	性能要求不高，且写多读少的场景。 默认值为 WriteBack。	
config.path	创建卷时，指定存储卷数据的数据目录（仅单机版支持）。 如果创建卷时不指定数据目录，使用服务器设置的默认数据目录。	否

### ● 响应结果

名称	描述
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips。 类型：数组
iqn	iSCSI Target IQN。
status	卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● Active：主 Target。</li> <li>● Standby：热备 Target。</li> <li>● Unavailable：冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
serverId	服务器 ID。

### ● 请求示例 1

单机版：创建卷 lun8。

POST /rest/v1/block/lun HTTP/1.1 Date: Wed, 18 Oct 2023 02:56:09 GMT Content-Type: application/json; charset=utf-8 Authorization: HBlock <i>userName:signature</i>
---

```
Content-Length: 210
Host:192.168.0.69:1443

{
  "lunName": "lun8",
  "targetName": "target8",
  "capacity": 500,
  "config": {
    "sectorSize": 512,
    "writePolicy": "WriteThrough",
    "path": "/mnt/storage01"
  }
}
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: c79736f32fe249cc8c40af6d90d3fd68
Connection: keep-alive
Content-Length: 158
Date: Wed, 18 Oct 2023 02:56:09 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target8.2",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.202",
            "port": 3260
          }
        ]
      }
    ],
  }
}
```

```
        "serverId": "hblock_1"
      }
    ]
  }
}
```

### ● 请求示例 2

集群版：创建卷 lun8。

```
POST /rest/v1/block/lun HTTP/1.1
Date: Thu, 22 Feb 2024 07:42:54 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 316
Host:192.168.0.192:1443

{
  "lunName": "lun8",
  "targetName": "target8",
  "capacity": 500,
  "config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 3,
    "ECfragmentSize": 1024,
    "sectorSize": 512,
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteThrough"
  }
}
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 3d4faf99cb9447bfaa337fd2a63187eb
Connection: keep-alive
Content-Length: 296
```

```
Date: Thu, 22 Feb 2024 07:42:54 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target8.38",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.102",
            "port": 3260
          }
        ],
        "serverId": "hblock_3"
      },
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target8.39",
        "status": "Unavailable",
        "ips": [
          {
            "ip": "192.168.0.192",
            "port": 3260
          }
        ],
        "serverId": "hblock_2"
      }
    ]
  }
}
```

## 4.3.2 启用卷

此操作用来启用卷。

- 请求语法

```
PUT /rest/v1/block/lun/LunName/enable HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

- 请求参数

参数	描述	是否必须
lunName	要启用的卷名称。 类型：字符串 取值：长度范围是 1~16，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头。	是

- 请求示例

启用卷 lun02a。

```
PUT /rest/v1/block/lun/lun02a/enable HTTP/1.1
Date: Fri, 29 Jul 2022 06:51:31 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
Content-Length: 0
```

- 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: d38ccce65daf426eb04e294c86beb5a1
Connection: keep-alive
Date: Fri, 29 Jul 2022 06:51:31 GMT
Server: HBlock
```

### 4.3.3 禁用卷

此操作用来禁用卷。

**注意：**如果禁用卷的编号是 0（卷的编号可以通过[查询指定卷](#)获取），禁用后：

- 如果客户端保持与该卷所属 iSCSI Target 的连接，则客户端可以继续正常访问该 iSCSI Target 下的其他卷。
- 如果客户端与该卷所属 iSCSI Target 断开连接后再重新连接，可能会导致无法再正常访问该 iSCSI Target 下的其他卷，如出现此情况，将该卷启用后即可重新正常访问。

#### ● 请求语法

```
PUT /rest/v1/block/lun/LunName/disable HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

参数	描述	是否必须
lunName	要禁用卷的卷名称。 类型：字符串 取值：长度范围是 1~16，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头。	是

#### ● 请求示例 1

禁用卷 lun02a，卷编号为 0。

```
PUT /rest/v1/block/lun/lun02a/disable HTTP/1.1
Date: Fri, 29 Jul 2022 07:12:46 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

```
Content-Length: 0
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 79c24ffcf1f84c0c8a6a2250cc8ffb96
Connection: keep-alive
Content-Type: application/json;charset=utf-8
Content-Length: 151
Date: Fri, 29 Jul 2022 07:12:46 GMT
Server: HBlock

{
  "data": {
    "message": "Disable LUN lun02a successfully. This is LUN 0, if the client
reconnects, it may not be able to access other LUNs of the target."
  }
}
```

- 请求示例 2

禁用卷 lun02b。

```
PUT /rest/v1/block/lun/lun02b/disable HTTP/1.1
Date: Fri, 29 Jul 2022 07:18:22 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
Content-Length: 0
```

- 响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: cda3a9d5ba424235803438a448294571
Connection: keep-alive
Date: Fri, 29 Jul 2022 07:18:23 GMT
Server: HBlock
```

### 4.3.4 删除卷

此操作用来删除卷。

- 请求语法

```
DELETE /rest/v1/block/lun/LunName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

- 请求参数

参数	描述	是否必须
lunName	要删除的卷名称。 类型：字符串 取值：1~16位字符串，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头。	是

- 请求示例

删除卷名称为 testLun1 的卷。

```
DELETE /rest/v1/block/lun/testLun1 HTTP/1.1
Date: Tue, 15 Mar 2022 02:40:17 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 75dd09b8f97941d98b74ac15b4b5c855
Connection: keep-alive
Content-Length: 0
Date: Tue, 15 Mar 2022 02:40:17 GMT
Server: HBlock
```

### 4.3.5 扩容卷

此操作用来扩容卷。

**注意：**卷扩容，除了在服务器端进行扩容外，还需要在客户端进行操作，详细操作可以参考命令行手册“4.6.5 扩容卷”章节。

- 请求语法

```
PUT /rest/v1/block/lun/LunName/expand HTTP/1.1
Date: date
Host: ip:port
Authorization:authorization

{
  "capacity": capacity
}
```

- 请求参数

参数	描述	是否必须
lunName	卷名称。 类型：字符串 取值：长度范围是 1~16，只能由字母、数字和短横线 (-) 组成，字母区分大小写，且仅支持以字母或数字开头。	是
capacity	卷要扩容后的容量。 类型：整型 取值：[1,1048576]，单位是 GiB。	是

- 请求示例

将卷 lunName1 扩容至 200000 GiB。

```
PUT /rest/v1/block/lun/lunName1/expand HTTP/1.1
Date: Fri, 29 Jul 2022 06:43:11 GMT
```

```
Content-Type: application/json; charset=utf-8
```

```
Authorization: HBlock userName:signature
```

```
Content-Length: 26
```

```
Host: 192.168.0.121:1443
```

```
{  
  "capacity":200000  
}
```

- 响应示例

```
HTTP/1.1 204 No Content
```

```
x-hblock-request-id: 49ed1b92c7e74137ae8d5572cb1a0aa1
```

```
Connection: keep-alive
```

```
Date: Fri, 29 Jul 2022 06:43:11 GMT
```

```
Server: HBlock
```

### 4.3.6 修改卷

此操作用来修改卷。

说明：卷处于启用或禁用状态时可以修改；卷处于“删除中”，不能修改。

#### ● 请求语法

```
PUT /rest/v1/block/lun/LunName HTTP/1.1
```

```
Date: date
```

```
Host: ip:port
```

```
Authorization: authorization
```

```
{
  "config":{
    "minReplica": minReplica,
    "writePolicy": writePolicy
  }
}
```

#### ● 请求参数

参数	描述	是否必须
lunName	卷名称。 类型：字符串 取值：长度范围是 1~16，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头	是
config.minReplica	最小副本数（仅集群版支持）。 对于副本模式的卷，假设卷副本数为 X，最小副本数为 Y（Y 必须≤X），该卷每次写入时，至少 Y 份数据写入成功，才视为本次写入成功。对于 EC N+M 模式的卷，假设该卷最小副本数设置为 Y（必须满足	否

	<p><math>N \leq Y \leq N+M</math>), 必须满足总和至少为 Y 的数据块和校验块写入成功, 才视为本次写入成功。</p> <p>类型: 整型</p> <p>取值: 对于副本卷, 取值范围是[1, N], N 为副本模式卷的副本数, 默认值为 1。对于 EC 卷, 取值范围是[N, N+M], 默认值为 N。</p>	
config.writePolicy	<p>卷的写策略。</p> <p>类型: 枚举</p> <p>取值:</p> <ul style="list-style-type: none"> <li>● <b>WriteBack:</b> 回写, 指数据写入到内存后即返回客户端成功, 之后再异步写入磁盘。适用于对性能要求较高, 稳定性要求不高的场景。</li> <li>● <b>WriteThrough:</b> 透写, 指数据同时写入内存和磁盘, 并在都写成功后再返回客户端成功。适用于稳定性要求较高, 写性能要求不高, 且最近写入的数据会较快被读取的场景。</li> <li>● <b>WriteAround:</b> 绕写, 指数据直接写到磁盘, 不写入内存。适用于稳定性要求较高, 性能要求不高, 且写多读少的场景。</li> </ul>	否

● 请求示例 1

修改卷 lun8 的写策略为 WriteThrough。

```
PUT /rest/v1/block/lun/lun8 HTTP/1.1
Date: Fri, 29 Jul 2022 07:29:17 GMT
Host: 192.168.0.121:1443
Content-Length: 53
Authorization: HBlock userName:signature

{
```

```
"config":{
  "writePolicy":"WriteThrough"
}
}
```

- 响应示例 1

```
HTTP/1.1 204 No Content
Date: Fri, 29 Jul 2022 07:29:18 GMT
Connection: keep-alive
x-hblock-request-id: 140a25c99a9a4efc87e561b44b595926
Server: HBlock
```

- 请求示例 2

修改卷的最小副本数为 3。

```
PUT /rest/v1/block/lun/lun01a HTTP/1.1
Date: Sat, 7 Oct 2023 07:58:35 GMT
Host: 192.168.0.117:1443
Content-Length: 53
Authorization: HBlock userName:signature

{
  "config": {
    "minReplica": 3
  }
}
```

- 响应示例 2

```
HTTP/1.1 204 No Content
Date: Sat, 7 Oct 2023 07:58:35 GMT
Connection: keep-alive
x-hblock-request-id: 437f52eff9fe44a6ad9aae796bed55e
Server: HBlock
```

### 4.3.7 触发卷对应 Target 的主备切换（集群版适用）

此操作用来触发卷对应 Target 的主备切换。

**注意：**执行此操作后，客户端不需要进行任何操作。

#### ● 请求语法

```
PUT /rest/v1/block/lun/LunName/switch HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

#### ● 请求参数

参数	描述	是否必须
lunName	卷名称。 类型：字符串 取值：长度范围是 1~16，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头。	是

#### ● 响应结果

名称	描述
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips。 类型：数组
iqn	Target IQN。
status	卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● Active: 主 Target。</li> <li>● Standby: 热备 Target。</li> <li>● Unavailable: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。

	类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。

- 请求示例

对卷 lun01a 进行切换。

```
PUT /rest/v1/block/lun/lun01a/switch HTTP/1.1
Date: Fri, 10 Feb 2023 09:42:01 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.117:1443
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 10d4e743ef08473fbe41cc61cd933844
Connection: keep-alive
Content-Length: 296
Date: Fri, 10 Feb 2023 09:42:01 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",
        "status": "Unavailable",
        "ips": [
          {
            "ip": "192.168.0.117",
            "port": 3260
          }
        ]
      }
    ],
    "serverId": "hblock_1"
  }
}
```

```
    },  
    {  
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",  
      "status": "Active",  
      "ips": [  
        {  
          "ip": "192.168.0.192",  
          "port": 3260  
        }  
      ],  
      "serverId": "hblock_3"  
    }  
  ]  
}
```

### 4.3.8 批量查询卷

此操作用来批量查询卷。可以通过设置过滤条件和 `range` 范围进行查询。如果未设置过滤条件和 `range` 范围，则查询所有的卷。

#### ● 请求语法

```
GET /rest/v1/block/lun?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

#### ● 请求参数

参数	描述	是否必须
<code>filter</code>	查询的过滤条件。可以选择多个查询，如果是或的关系，使用“or”将查询条件分隔开；如果是与的关系，使用“and”将查询条件分隔开。 支持的过滤条件类型包括： <ul style="list-style-type: none"> <li>● <code>lunName</code>: 卷名称。</li> <li>● <code>status</code>: 卷的状态。</li> <li>● <code>capacity</code>: 卷容量。</li> <li>● <code>iqn</code>: Target 的识别。</li> </ul>	否
<code>range</code>	查询卷的范围，格式为 <code>i-j</code> ， <code>i</code> 和 <code>j</code> 为正整数， $0 < j-i < 1000$ 。按照卷名称进行排序，返回第 <code>i</code> 到第 <code>j</code> 的卷。 如果输入的查询范围超过卷的总个数，返回的结果为空，如果未指定 <code>range</code> ，则返回所有符合条件的卷。	否

#### ● 响应结果

名称	描述
----	----

lunName	卷名称。
status	卷的状态： <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 卷处于启用状态。</li> <li>● <b>Disabled</b>: 卷处于禁用状态。</li> <li>● <b>Deleting</b>: 卷正在删除中。</li> </ul>
capacity	卷容量，单位是 GiB。
createTime	卷创建时间。unix 时间戳（UTC），精确到毫秒。
storageMode	存储类型： <p>Local: 本地模式，数据全部保留在本地。</p>
iSCSITargets	iSCSI Target 属性集合。包含: iqn、status、ips、serverId。 类型: 数组
iqn	iSCSI Target IQN。
status	卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● <b>Active</b>: 主 Target。</li> <li>● <b>Standby</b>: 热备 Target。</li> <li>● <b>Offline</b>: 离线。</li> <li>● <b>Unavailable</b>: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含: ip、port。 类型: 数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
serverId	IQN 所在的服务器 ID。
config.localStorageClass	卷冗余模式（仅集群版支持）： <ul style="list-style-type: none"> <li>● <b>single-copy</b>: 单副本。</li> <li>● <b>2-copy</b>: 两副本。</li> <li>● <b>3-copy</b>: 三副本。</li> </ul>

	<ul style="list-style-type: none"> <li>● EC <math>N+M</math>。</li> </ul>
config.minReplica	卷的最小副本数（仅集群版支持）。
config.ECfragmentSize	分片大小。如果冗余模式为 EC $N+M$ ，则会返回此项。
config.sectorSize	扇区大小： <ul style="list-style-type: none"> <li>● 512。</li> <li>● 4096。</li> </ul>
config.highAvailability	高可用模式类型（仅集群版支持）： <ul style="list-style-type: none"> <li>● ActiveStandby: 主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● Disabled: 禁用，该卷关联对应 Target 下的 1 个 IQN。</li> </ul>
config.writePolicy	卷的写策略： <ul style="list-style-type: none"> <li>● WriteBack: 回写，指数据写入到内存后即返回客户端成功，之后再异步写入磁盘。</li> <li>● WriteThrough: 透写，指数据同时写入内存和磁盘，并在都写成功后再返回客户端成功。</li> <li>● WriteAround: 绕写，指数据直接写到磁盘，不写入内存。</li> </ul>
config.path	数据目录（仅单机版支持）。
number	卷编号。
WWID	卷的唯一标识符。

- 请求示例 1

单机版：查询所有卷。

```
GET /rest/v1/block/lun HTTP/1.1
Date: Thu, 22 Feb 2024 07:54:13 GMT
Authorization: userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: b4e8d60565134da2b39fb68d3b045c2b
Connection: keep-alive
Content-Length: 1104
Date: Thu, 22 Feb 2024 07:54:13 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "luna1",
        "status": "Enabled",
        "capacity": 20,
        "storageMode": "Local",
        "createTime": 1706514173255,
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta1.1",
            "status": "Active",
            "ips": [
              {
                "ip": "192.168.0.32",
                "port": 3260
              }
            ],
            "serverId": "hblock_1"
          }
        ],
        "config": {
          "sectorSize": 4096,
          "writePolicy": "WriteBack",

```

```
        "path": "/mnt/stor"
    },
    "number": 0,
    "WWID": "330000000336eb649"
},
{
    "lunName": "luna2",
    "status": "Enabled",
    "capacity": 25,
    "storageMode": "Local",
    "createTime": 1706514191085,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta1.1",
            "status": "Active",
            "ips": [
                {
                    "ip": "192.168.0.32",
                    "port": 3260
                }
            ],
            "serverId": "hblock_1"
        }
    ],
    "config": {
        "sectorSize": 4096,
        "writePolicy": "WriteBack",
        "path": "/mnt/stor"
    },
    "number": 1,
    "WWID": "33ffffffffb9c0da47"
},
{
    "lunName": "lunb1",
    "status": "Enabled",
```

```

        "capacity": 22,
        "storageMode": "Local",
        "createTime": 1706838591621,
        "iSCSITargets": [
            {
                "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetb.3",
                "status": "Active",
                "ips": [
                    {
                        "ip": "192.168.0.32",
                        "port": 3260
                    }
                ],
                "serverId": "hblock_1"
            }
        ],
        "config": {
            "sectorSize": 4096,
            "writePolicy": "WriteBack",
            "path": "/mnt/stor"
        },
        "number": 0,
        "WWID": "3300000003294c4d1"
    }
}
}
}
    
```

● 请求示例 2

集群版：查询卷名称包含 lun 的卷，并且只列出第 1 个到第 3 个查询结果。

```

GET /rest/v1/block/lun?filter=lunName:lun&range=1-3 HTTP/1.1
Date: Thu, 22 Feb 2024 07:52:16 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
    
```

- 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: c4b34d27b2c4439190cc63f7cf2b7bd0
Connection: keep-alive
Content-Length: 1749
Date: Thu, 22 Feb 2024 07:52:16 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "lun01a",
        "status": "Enabled",
        "capacity": 20,
        "storageMode": "Local",
        "createTime": 1706514101509,
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
            "status": "Active",
            "ips": [
              {
                "ip": "192.168.0.110",
                "port": 3260
              }
            ],
            "serverId": "hblock_1"
          },
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
            "status": "Standby",
            "ips": [
```

```
        {
            "ip": "192.168.0.102",
            "port": 3260
        }
    ],
    "serverId": "hblock_3"
}
],
"config": {
    "localStorageClass": "EC 2+1",
    "minReplica": 2,
    "sectorSize": 4096,
    "highAvailability": "ActiveStandby",
    "writePolicy": "WriteBack",
    "ECfragmentSize": 16
},
"number": 0,
"WWID": "3300000004783ef0b"
},
{
    "lunName": "lun01b",
    "status": "Enabled",
    "capacity": 30,
    "storageMode": "Local",
    "createTime": 1706514206007,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
            "status": "Active",
            "ips": [
                {
                    "ip": "192.168.0.110",
                    "port": 3260
                }
            ]
        }
    ],
}
```

```
        "serverId": "hblock_1"
      },
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
        "status": "Standby",
        "ips": [
          {
            "ip": "192.168.0.102",
            "port": 3260
          }
        ],
        "serverId": "hblock_3"
      }
    ],
    "config": {
      "localStorageClass": "EC 2+1",
      "minReplica": 2,
      "sectorSize": 4096,
      "highAvailability": "ActiveStandby",
      "writePolicy": "WriteBack",
      "ECfragmentSize": 16
    },
    "number": 1,
    "WWID": "330000001f0655ca"
  },
  {
    "lunName": "lun02a",
    "status": "Enabled",
    "capacity": 30,
    "storageMode": "Local",
    "createTime": 1706514891549,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.5",
        "status": "Active",
```

```
        "ips": [  
            {  
                "ip": "192.168.0.192",  
                "port": 3260  
            }  
        ],  
        "serverId": "hblock_2"  
    },  
    {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.6",  
        "status": "Standby",  
        "ips": [  
            {  
                "ip": "192.168.0.110",  
                "port": 3260  
            }  
        ],  
        "serverId": "hblock_1"  
    }  
],  
"config": {  
    "localStorageClass": "2-copy",  
    "minReplica": 1,  
    "sectorSize": 4096,  
    "highAvailability": "ActiveStandby",  
    "writePolicy": "WriteBack",  
    "ECfragmentSize": 4  
},  
"number": 0,  
"WWID": "3300000001b2644ee"  
}  
]  
}
```

### 4.3.9 查询指定卷

此操作用来查询指定卷。

#### ● 请求语法

```
GET /rest/v1/block/lun/LunName HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

参数	描述	是否必须
lunName	卷名称。 <b>说明：</b> 如果不填写此项，会返回所有卷信息。 类型：字符串 取值： 长度范围是 1~16，只能由字母、数字和短横线（-）组成，字母区分大小写，且仅支持以字母或数字开头。	否

#### ● 响应结果

名称	描述
lunName	卷名称。
targetName	Target 名称。
status	卷的状态： <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 卷处于启用状态。</li> <li>● <b>Disabled:</b> 卷处于禁用状态。</li> <li>● <b>Deleting:</b> 卷正在删除中。</li> </ul>
capacity	卷容量，单位是 GiB。
storageMode	存储类型：

	<b>Local</b> : 本地模式, 数据全部保留在本地。
<b>createTime</b>	卷创建时间。 <b>unix</b> 时间戳 (UTC), 精确到毫秒。
<b>iSCSITargets</b>	iSCSI Target 属性集合。包含: <b>iqn</b> 、 <b>status</b> 、 <b>ips</b> 、 <b>targetPortalIP</b> 、 <b>serverId</b> 。 类型: 数组
<b>iqn</b>	Target IQN。
<b>status</b>	卷对应的 Target 的状态: <ul style="list-style-type: none"> <li>● <b>Active</b>: 主 Target。</li> <li>● <b>Standby</b>: 热备 Target。</li> <li>● <b>Offline</b>: 离线。</li> <li>● <b>Unavailable</b>: 冷备 Target。</li> </ul>
<b>ips</b>	iSCSI Target 的 IP 属性集合, 包含: <b>ip</b> 、 <b>port</b> 。 类型: 数组
<b>ip</b>	iSCSI Target 的 IP。
<b>port</b>	iSCSI Target 的端口号。
<b>targetPortalIP.ips</b>	iSCSI 目标门户的属性集合, 包括: <b>ip</b> 、 <b>port</b> 。 类型: 数组
<b>ip</b>	iSCSI 目标门户的 IP 地址。
<b>port</b>	iSCSI 目标门户的端口号。
<b>targetPortalIP.status</b>	iSCSI 目标门户的状态: <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 启用状态。</li> <li>● <b>Disabled</b>: 禁用状态。</li> </ul>
<b>serverId</b>	IQN 所在的服务器 ID。
<b>config.localStorageClass</b>	卷冗余模式 (仅集群版支持): <ul style="list-style-type: none"> <li>● <b>single-copy</b>: 单副本。</li> <li>● <b>2-copy</b>: 两副本。</li> <li>● <b>3-copy</b>: 三副本。</li> </ul>

	<ul style="list-style-type: none"> <li>● EC N+M。</li> </ul>
config.minReplica	卷的最小副本数。
config.ECfragmentSize	分片大小。如果冗余模式为 EC N+M，则会返回此项。
config.sectorSize	扇区大小，单位为字节。
config.highAvailability	高可用模式类型（仅集群版支持）： <ul style="list-style-type: none"> <li>● <b>ActiveStandby</b>: 主备，该卷关联对应 Target 下的所有 IQN。</li> <li>● <b>Disabled</b>: 禁用，该卷关联对应 Target 下的 1 个 IQN。</li> </ul>
config.writePolicy	卷的写策略： <ul style="list-style-type: none"> <li>● <b>WriteBack</b>: 回写，指数据写入到内存后即返回客户端成功，之后再异步写入磁盘。</li> <li>● <b>WriteThrough</b>: 透写，指数据同时写入内存和磁盘，并在都写成功后再返回客户端成功。</li> <li>● <b>WriteAround</b>: 绕写，指数据直接写到磁盘，不写入内存。</li> </ul>
config.path	数据目录（仅单机版支持）。
number	卷编号。LUN 在 Target 下的编号，由存储系统分配，对应客户端挂载存储设备时设备地址中的 LUN ID。如果 Target 下只有一个 LUN，LUN 的编号一般为 0。
uuid	卷的唯一识别码。
WWID	卷的唯一标识符。

- 请求示例 1

单机版：查询卷 luna1

```
GET /rest/v1/block/lun/luna1 HTTP/1.1
Date: Thu, 22 Feb 2024 07:56:32 GMT
```

```
Authorization: HBlock userName:signature
```

```
Host: 192.168.0.32:1443
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
```

```
x-hblock-request-id: 2b2a736d53c040febdde70f0931fe2dc
```

```
Connection: keep-alive
```

```
Content-Length: 449
```

```
Date: Thu, 22 Feb 2024 07:56:32 GMT
```

```
Content-Type: application/json;charset=utf-8
```

```
Server: HBlock
```

```
{
  "data": {
    "lunName": "luna1",
    "targetName": "targeta1",
    "status": "Enabled",
    "capacity": 20,
    "storageMode": "Local",
    "createTime": 1706514173255,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta1.1",
        "status": "Active",
        "ips": [
          {
            "ip": "192.168.0.32",
            "port": 3260
          }
        ],
        "serverId": "hblock_1"
      }
    ],
    "config": {
      "sectorSize": 4096,

```

```
    "writePolicy": "WriteBack",
    "path": "/mnt/stor"
  },
  "number": 0,
  "uuid": "lun-uuid-28b22092-2a06-4065-ab68-399865e9ea72",
  "WWID": "330000000336eb649"
}
}
```

### ● 请求示例 2

集群版：查询卷 lun01a。

```
GET /rest/v1/block/lun/lun01a HTTP/1.1
Date: Thu, 22 Feb 2024 08:04:11 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 31a1a6bbd7874999b6bb87275c67e4d0
Connection: keep-alive
Content-Length: 664
Date: Thu, 22 Feb 2024 08:04:11 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "lunName": "lun01a",
    "targetName": "target01",
    "status": "Enabled",
    "capacity": 20,
    "storageMode": "Local",
    "createTime": 1706514101509,
    "iSCSITargets": [
```

```
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.2",
  "status": "Active",
  "ips": [
    {
      "ip": "192.168.0.110",
      "port": 3260
    }
  ],
  "serverId": "hblock_1"
},
{
  "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.1",
  "status": "Standby",
  "ips": [
    {
      "ip": "192.168.0.102",
      "port": 3260
    }
  ],
  "serverId": "hblock_3"
}
],
"config": {
  "localStorageClass": "EC 2+1",
  "minReplica": 2,
  "sectorSize": 4096,
  "highAvailability": "ActiveStandby",
  "writePolicy": "WriteBack",
  "ECfragmentSize": 16
},
"number": 0,
"uuid": "lun-uuid-ce84e7e9-95b9-4a32-86f4-589822476a33",
"WWID": "3300000004783ef0b"
}
```

```
} 
```

### 4.3.10 错误码列表

HTTP status	错误码	错误信息	描述
400	DisableDeletingLUN	Failed to disable LUN because the LUN <i>lunName</i> is deleting.	禁用卷失败，因为该卷正在删除。
400	EnableDeletingLUN	Failed to enable LUN because the LUN <i>lunName</i> is deleting.	启用卷失败，因为该卷正在删除。
400	ExceedMaxCapacity	Value <i>value</i> at 'capacity' failed to satisfy constraint: Argument must be less than <i>value</i> .	存储容量非法。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed value.	超出了系统允许的最大个数，无法执行该操作。
400	HighAvailabilityNotAllowed	The 'high availability' is invalid for standalone mode of HBlock.	单机版本 HBlock 不支持设置卷的高可用模式。
400	InvalidFilterEnum	Value <i>value</i> at 'filter' failed to satisfy constraint: Argument can only contain: [lunName, status, capacity, targetName].	Filter 参数不正确。
400	InvalidHighAvailability	Value <i>value</i> at 'high availability' failed to satisfy constraint:	高可用模式取值无效。

		Argument must satisfy enum value set: [Disabled, ActiveStandby].	
400	InvalidLUNName	Value <i>value</i> at 'lun name' failed to satisfy constraint: Argument must contain only letters, digits, or hyphens (-), begin with a letter or digit, and does not exceed 16 characters.	卷名称不正确。
400	InvalidMinReplicaNumber	Argument 'min replica' failed to satisfy constraint: argument range is [1, copy number] for replica mode and [N, N+M] for EC N+M mode.	最小副本数取值范围为：副本卷为[1, 副本数]，EC 卷为[N, N+M]。
400	InvalidPathStatus	The path <i>path</i> status is ' <i>status</i> ', the request is invalid.	数据目录状态不正确，无法执行操作。
400	InvalidPositiveInteger	Value <i>value</i> at ' <i>number</i> ' failed to satisfy constraint: Argument must be positive integer.	值必须为正整数。
400	InvalidRange	Value <i>value</i> at 'range' failed to satisfy constraint: Argument must satisfy pattern 'i-j', where i, j are	range 参数不正确。

		positive integers, $i$ is less than $j$ , and $j-i < 1000$ .	
400	InvalidSectorSize	Value <i>value</i> at 'sector size' failed to satisfy constraint: Argument must satisfy enum value set: [512, 4096].	卷的块大小不在枚举范围内。
400	InvalidStorageClass	Value <i>value</i> at 'local storage class' failed to satisfy constraint: Argument must satisfy enum value set: [single-copy, 2-copy, 3-copy, EC $N+M$ ], $N$ and $M$ are positive integers, $N > M$ and $N + M \leq 128$ .	副本模式与当前实际不匹配，服务器个数不够。
400	InvalidTargetName	Value <i>value</i> at 'target name' failed to satisfy constraint: Argument must contain only lowercase letters, digits, dots(.), or hyphens (-), begin with a lowercase letter or digit, and does not exceed 16 characters.	Target 名称不正确。
400	InvalidWritePolicy	Value <i>value</i> at 'write policy' failed to satisfy	写策略非法。

		constraint: Argument must satisfy enum value set: [WriteBack, WriteThrough, WriteAround].	
400	LessThanCurrentCapacity	Value <i>value</i> at 'capacity' failed to satisfy constraint: Argument must be greater than <i>value</i> .	扩容卷时新容量必须大于当前容量。
400	LessThanMinCapacity	Value <i>value</i> at 'capacity' failed to satisfy constraint: Argument must be greater than or equal to 1 GiB.	容量不能小于 1 GiB。
400	LUNDuplicatedCatalog	Duplicated with the existing LUN catalog.	与已有卷的数据目录重复。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	StorageClassNotAllowed	The 'local storage class' is invalid for standalone mode of HBlock.	单机版 HBlock 不支持设置副本模式。
400	SwitchLUNNotAllowed	Failed to switch LUN. The LUN does not support high availability.	单控卷不支持卷的主备切换。
400	TooManyLUNs	The number of LUNs in a Target cannot exceed 256.	单个 Target 中的卷不能超过 256 个。
404	NoSuchLicense	The HBlock license does	没有软件许可证。

		not exist.	
404	NoSuchLUN	The LUN with name <i>lunName</i> does not exist.	卷不存在。
404	NoSuchPath	The path <i>path</i> does not exist in the HBlock.	路径不存在。
409	LUNAlreadyExists	The LUN with name <i>lunName</i> already exists.	卷已经存在。
409	DeleteLUNEnabled	Cannot delete it because the iSCSI LUN is enabled, please disable first.	不能删除正在启用的卷。
409	InvalidTargetIQNsStatus	The request is invalid. Ensure the number of Target IQNs whose status is neither unavailable nor offline is greater than or equal to 2 for the LUN <i>lunname,lunname,....</i>	操作失败，Target IQNs 状态异常，请检查后重试。

## 4.4 iSCSI Target 管理

### 4.4.1 创建 iSCSI Target

此操作用来创建 iSCSI Target。

说明：HBlock 上 Target IQN 的数量最多 32766 个。一个 Target 最多可以关联 256 个卷，但是一个卷只能被一个 Target 关联。

- 请求语法

#### 单机版

```
POST /rest/v1/block/target HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "targetName": targetName,
  "maxSessions": value,
  "chap":{
    "name": name,
    "password": password,
    "status": status
  }
}
```

#### 集群版

```
POST /rest/v1/block/target HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization
```

```

{
  "targetName": targetName,
  "maxSessions": value,
  "chap":{
    "name": name,
    "password": password,
    "status": status
  },
  "serverNumbers": value,
  "server":["server_ID",...]
}
    
```

● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点（.）和短横线（-）组成，且仅支持以字母或数字开头。	是
maxSessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。 类型：整型 取值：取值范围是[0, 1024]，默认值为 1。0 表示客户端无法发现该 Target。 <b>注意：</b> 如果多个客户端连接同一 Target IQN，客户端可以同时读，但不能同写。	否
chap.name	CHAP 名称。 类型：字符串 取值：长度范围是 3~64，可以由字母、数字、句点（.）、短横线（-）、下划线（_）和冒号（:）组成，字母区分大小	否

	写，且仅支持以字母或数字开头。	
chap.password	CHAP 密码。 类型：字符串 取值：长度范围是 12~16，可以由字母、数字、下划线（_）组成，字母区分大小写。	否
chap.status	是否启用 CHAP 认证。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用 CHAP 认证。</li> <li>● Disabled: 禁用 CHAP 认证。</li> </ul> 注意：如果启用 CHAP 认证，name 和 password 必须设置。	是
serverNumbers	Target 所在的服务器数量（仅集群版支持）。 类型：整型 取值：[2, n]，n 为集群内服务器的数量。默认值是 2。	否
server	指定 iSCSI Target 对应的服务器 ID（仅集群版支持）。 最多可以指定 n 个服务器 ID，以英文逗号隔开。n 为集群内服务器的数量。	否

### ● 响应结果

名称	描述
iSCSITargets	iSCSI Target 属性集合。包含：iqn、ips、 类型：数组
iqn	iSCSI Target IQN。
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。

port	iSCSI Target 的端口号。
------	--------------------

- 请求示例 1

单机版：创建一个名称为 **targete** 的 Target。

```
POST /rest/v1/block/target HTTP/1.1
Date: Tue, 14 Feb 2023 07:58:44 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 155
Host: 192.168.0.32:1443

{
  "targetName": "targete",
  "chap": {
    "name": "test",
    "password": "123456789012",
    "status": "Enabled"
  }
}
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 8c2a6566e6d84b5e82ac26387c6a50ac
Connection: keep-alive
Content-Length: 217
Date: Tue, 14 Feb 2023 07:58:44 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
```

```

        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targete.6",
        "ips": [
            {
                "ip": "192.168.0.32",
                "port": 3260
            }
        ]
    }
}

```

- 请求示例 2

集群版创建一个名称为 **target06** 的 Target。

```

POST /rest/v1/block/target HTTP/1.1
Date: Thu, 28 Sep 2023 07:39:52 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 252
Host: 192.168.0.192:1443

{
  "targetName": "target06",
  "maxSessions": 6,
  "chap": {
    "name": "test",
    "password": "123456789012",
    "status": "Enabled"
  },
  "serverNumbers": 3,
  "server": ["hblock_1", "hblock_2"]
}

```

- 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: e1e127f0345c4b9ebc36d168998901a9
Connection: keep-alive
Content-Length: 303
Date: Thu, 28 Sep 2023 07:39:52 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target06.6",
        "ips": [
          {
            "ip": "192.168.0.110",
            "port": 3260
          }
        ]
      },
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target06.7",
        "ips": [
          {
            "ip": "192.168.0.117",
            "port": 3260
          }
        ]
      },
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target06.8",
        "ips": [
          {
            "ip": "192.168.0.102",
            "port": 3260
          }
        ]
      }
    ]
  }
}
```

```
}  
  }  
    ]  
  }  
  ]  
}  
}
```

## 4.4.2 修改 iSCSI Target 的 CHAP 认证

此操作用来修改 iSCSI Target 的 CHAP 认证。

### ● 请求语法

```
PUT /rest/v1/block/target/targetName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "chap": {
    "name": name,
    "password": password,
    "status": status
  }
}
```

### ● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是
chap.name	CHAP 名称。 类型：字符串 取值：长度范围是 3~64，可以由字母、数字、句点 (.)、短横线 (-)、下划线 (_) 和冒号 (:) 组成，字母区分大小写，且仅支持以字母或数字开头。	否

chap.password	CHAP 密码。 类型：字符串 取值：长度范围是 12~16，可以由字母、数字、下划线（_）组成，字母区分大小写。	否
chap.status	是否启用 CHAP 认证。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用 CHAP 认证。</li> <li>● Disabled: 禁用 CHAP 认证。</li> </ul> 注意：如果是首次启用 CHAP 认证，name 和 password 必须设置。	是

● 请求示例 1

修改 target2：启用 CHAP 认证，并设置 CHAP 认证的名称和密码。

```

PUT / rest/v1/block/target/target2 HTTP/1.1
Date: Fri, 29 Jul 2022 06:17:27 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 85
Host: 192.168.0.121:1443

{
  "chap":{
    "name":"test",
    "password":"*****",
    "status":"Enabled"
  }
}
    
```

● 响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: b18c4215a42a460789fb54879acfa3e3
Connection: keep-alive
Date: Fri, 29 Jul 2022 06:17:27 GMT
Server: HBlock
```

- 请求示例 2

修改 target2: 禁止 CHAP 认证, 但不删除先前 CHAP 认证的名称和密码。

```
PUT /rest/v1/block/target/target2 HTTP/1.1
Date: Fri, 29 Jul 2022 06:17:53 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 60
Host: 192.168.0.121:1443
{
  "chap":{
    "status":"Disabled"
  }
}
```

- 响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: 850b187b2d844b928998a190b8a034f3
Connection: keep-alive
Date: Fri, 29 Jul 2022 06:17:53 GMT
Server: HBlock
```

### 4.4.3 迁移 iSCSI Target（集群版适用）

此操作用来迁移 iSCSI Target，修改 iSCSI Target 对应的服务器。

说明：可以只迁移 iSCSI Target 对应的一个服务器，也可以同时迁移对应的多个服务器。

注意：

- 如果要迁移的 iSCSI Target 在源服务器上存在针对该 iSCSI Target 的连接，需要先断开该 iSCSI Target 对应的连接才能执行迁移。
- 执行迁移 iSCSI Target 之前，需要保证集群处于 working 状态，同时源和目的服务器需要处于正常已连接状态。

#### ● 请求语法

```
PUT /rest/v1/block/target/targetName/migrate HTTP/1.1
```

```
Date: date
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: length
```

```
Host: ip:port
```

```
Authorization: authorization
```

```
{
  "server": [
    {
      "source": source_server_ID,
      "dest": dest_server_ID
    },
    .....
    {
      "source": source_server_ID,
      "dest": dest_server_ID
    }
  ]
}
```

## ● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是
source	iSCSI Target 的源服务器 ID。	是
dest	iSCSI Target 的目标服务器 ID。	是

## ● 响应结果

名称	描述
luns	卷的属性集合，包含：lunName、iSCSITargets、 类型：数组
lunName	关联卷的名称。
iSCSITargets	iSCSI Target 属性集合。包含：iqn、status、ips、 targetPortalIP。 类型：数组
iqn	iSCSI Target IQN。
status	卷对应的 iSCSI Target 的状态： <ul style="list-style-type: none"> <li>● Active: 主 Target。</li> <li>● Standby: 热备 Target。</li> <li>● Offline: 离线。</li> <li>● Unavailable: 冷备 Target。</li> </ul>
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。

targetPortalIP.ips	iSCSI 目标门户的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI 目标门户的 IP 地址。 说明：如果未配置，不返回该项。
port	iSCSI 目标门户的端口号。 说明：如果未配置，不返回该项。
targetPortalIP.status	iSCSI 目标门户的状态： <ul style="list-style-type: none"> <li>● Enabled: 启用状态。</li> <li>● Disabled: 禁用状态。</li> </ul>

● 请求示例

迁移 target01。

```

PUT /rest/v1/block/target/target01/migrate HTTP/1.1
Date: Tue, 14 Feb 2023 08:20:21 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 95
Host: 192.168.0.192:1443

{
  "server": [
    {
      "source": "hblock_2",
      "dest": "hblock_1"
    }
  ]
}
    
```

● 响应示例

```

HTTP/1.1 200 OK
x-hblock-request-id: 4126f0d91c734357a4268828a2939d8c
    
```

```
Connection: keep-alive
Content-Type: application/json; charset=utf-8
Content-Length: 424
Date: Tue, 14 Feb 2023 08:20:21 GMT
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "lun01a",
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",
            "status": "Unavailable",
            "ips": [
              {
                "ip": "192.168.0.117",
                "port": 3260
              }
            ],
            "targetPortalIP": {
              "ips": [
                {
                  "ip": "101.89.213.5",
                  "port": 3261
                }
              ]
            },
            "status": "Enabled"
          }
        ],
        {
          "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",
          "status": "Active",
          "ips": [
```

```
{
  [
    {
      "ip": "192.168.0.192",
      "port": 3260
    }
  ]
}
```

**说明：**卷关联的 iSCSI Target 处于 **Unavailable** 状态，表示 iSCSI Target 正在从一个服务器迁移到集群其他服务器上，迁移过程，该 iSCSI Target 处于 **Unavailable** 状态。迁移完成后，iSCSI Target 将变为 **Active** 或者 **Standby** 状态。

#### 4.4.4 修改 iSCSI Target 下每个 IQN 允许建立的最大会话数

此操作用来修改 iSCSI Target 下每个 IQN 允许建立的最大会话数。

##### ● 请求语法

```
PUT /rest/v1/block/target/targetName HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "maxSessions": value
}
```

##### ● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是
maxSessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。 类型：整型 取值：取值范围是 [0, 1024]，默认值为 1。0 表示客户端无法发现该 Target。 <b>说明：</b> 建议修改为比现有会话数大的数字，因为如果修改为比现有会话数小，可能会导致现有客户端连接断开之后，不能再次建立连接。 <b>注意：</b> 如果多个客户端连接同一 Target IQN，客户端可以同	否

	时读，但不能同写。	
--	-----------	--

- 请求示例 1

将 target06 下每个 IQN 允许建立的最大会话数修改为 11。

```
PUT / rest/v1/block/target/target06 HTTP/1.1
Date: Thu, 28 Sep 2023 07:53:46 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 28
Host: 192.168.0.117:1443

{
  "maxSessions": 11
}
```

- 响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: a0a69bafb19a4371a978fc4bcfee48ac
Connection: keep-alive
Date: Thu, 28 Sep 2023 07:53:46 GMT
Server: HBlock
```

## 4.4.5 删除 iSCSI Target

此操作用来删除 iSCSI Target。

**注意：**只有未关联任何卷的 iSCSI Target 才能被删除。

### ● 请求语法

```
DELETE /rest/v1/block/target/targetName HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

### ● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是

### ● 请求示例

删除名为 target2 的 Target。

```
DELETE /rest/v1/block/target/target2 HTTP/1.1
```

Date: Wed, 16 Mar 2022 01:40:22 GMT

Authorization: HBlock *userName:signature*

Host: 192.168.0.121:1443

### ● 响应示例

```
HTTP/1.1 204 No Content
```

x-hblock-request-id: c9ca51dddbbd4cafa077fa86180a8e51

Connection: keep-alive

Date: Wed, 16 Mar 2022 01:40:22 GMT

Server: HBlock

## 4.4.6 删除 CHAP

此操作用来删除 CHAP 名称和密码。

### ● 请求语法

```
DELETE /rest/v1/block/target/targetName/chap HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是

### ● 请求示例

删除 target2 的 CHAP 名称和密码。

```
DELETE /rest/v1/block/target/target2/chap HTTP/1.1
Date: Wed, 16 Mar 2022 01:25:42 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: aa3fa33cdeba42ffa888392f92500772
Connection: keep-alive
Date: Wed, 16 Mar 2022 01:25:43 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

## 4.4.7 批量查询 iSCSI Target

此操作用来批量查询 iSCSI Target。

### ● 请求语法

```
GET /rest/v1/block/target?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。可以选择多个查询，如果是或的关系，使用 or 将查询条件分隔开；如果是与的关系，使用 and 将查询条件分隔开。 支持的过滤条件类型包括： <ul style="list-style-type: none"> <li>● lunName: iSCSI Target 关联的卷名称。</li> <li>● targetName: iSCSI Target 名称。</li> <li>● iqn: iSCSI Target IQN。</li> </ul>	否
range	查询 iSCSI Target 的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。 按照 Target 名称进行排序，返回第 i 到第 j 的 Target。 如果输入的查询范围超过 iSCSI Target 的总个数，返回的结果为空， 如果未指定 range，则返回所有符合条件的 iSCSI Target。	否

### ● 响应结果

名称	描述
targetName	iSCSI Target 名称。
maxSessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。
iSCSITargets	iSCSI Target 属性集合。包含: iqn、ips、

	targetPortalIP、serverId。 类型：数组
iqn	iSCSI Target IQN。
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
targetPortalIP.ips	iSCSI 目标门户 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI 目标门户 IP。 说明：如果未配置，不返回该项。
port	iSCSI 目标门户端口号。 说明：如果未配置，不返回该项。
targetPortalIP.status	iSCSI 目标门户 IP 状态： <ul style="list-style-type: none"> <li>● Enabled：启用状态。</li> <li>● Disabled：停用状态。</li> </ul> 说明：如果未配置，不返回该项。
serverId	iSCSI Target 的 IQN 所在的服务器 ID。
chap.name	CHAP 认证名称。
chap.status	CHAP 认证状态： <ul style="list-style-type: none"> <li>● Enabled：启用 CHAP 认证。</li> <li>● Disabled：禁用 CHAP 认证。</li> </ul>
createTime	创建 iSCSI Target 的时间，unix 时间戳（UTC），精确到毫秒。
serverIds	iSCSI Target 对应的服务器 ID（仅集群版支持）。
num	Target 所在的服务器数量（仅集群版支持）。

- 请求示例 1

单机版：查询 iSCSI Target 名称中包含 **target**，且关联的卷名称包含 **lun** 的 iSCSI Target，且只列出第 1 个到第 100 个的查询结果。

```
GET /rest/v1/block/target?filter=targetName:target and lunName:lun&range=1-100 HTTP/1.1
Date: Thu, 28 Sep 2023 09:10:40 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: afa77ab80a234f31bc6e58d9512cae99
Connection: keep-alive
Content-Length: 637
Date: Thu, 28 Sep 2023 09:10:40 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "targets": [
      {
        "targetName": "targeta",
        "maxSessions": 1,
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.2",
            "ips": [
              {
                "ip": "192.168.0.32",
                "port": 3260
              }
            ],
            "targetPortalIP": {
              "ips": [
```

```
        {
            "ip": "10.1.1.10",
            "port": 20430
        }
    ],
    "status": "Disabled"
},
"serverId": "hblock_1"
}
],
"createTime": 1695891789350,
"num": 1
},
{
    "targetName": "targetc",
    "maxSessions": 3,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:targetc.1",
            "ips": [
                {
                    "ip": "192.168.0.32",
                    "port": 3260
                }
            ],
            "targetPortalIP": {
                "ips": [
                    {
                        "ip": "10.1.1.10",
                        "port": 20430
                    }
                ]
            },
            "status": "Disabled"
        },
        "serverId": "hblock_1"
    ]
}
```

```
    }
  ],
  "chap": {
    "name": "chap-test",
    "status": "Enabled"
  },
  "createTime": 1695806964763,
  "num": 1
}
]
}
}
```

### ● 请求示例 2

集群版：查询 iSCSI Target 名称中包含 **target0**，且关联的卷名称包含 **lun0** 的 iSCSI Target，且只列出第 1 个到第 100 个的查询结果。

```
GET /rest/v1/block/target?filter=targetName:target0 and lunName:lun0&range=1-100
HTTP/1.1
Date: Thu, 28 Sep 2023 08:46:43 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 2fe3309a0e7c4171bde34ad4923222d0
Connection: keep-alive
Content-Length: 992
Date: Thu, 28 Sep 2023 08:46:43 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
```

```
"targets": [  
  {  
    "targetName": "target01",  
    "maxSessions": 10,  
    "iSCSITargets": [  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",  
        "ips": [  
          {  
            "ip": "192.168.0.117",  
            "port": 3260  
          }  
        ],  
        "targetPortalIP": {  
          "ips": [  
            {  
              "ip": "10.0.1.10",  
              "port": 10011  
            }  
          ],  
          "status": "Disabled"  
        },  
        "serverId": "hbblock_1"  
      },  
      {  
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",  
        "ips": [  
          {  
            "ip": "192.168.0.102",  
            "port": 3260  
          }  
        ],  
        "serverId": "hbblock_2"  
      },  
      {  
        "targetName": "target02",  
        "maxSessions": 10,  
        "iSCSITargets": [  
          {  
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.3",  
            "ips": [  
              {  
                "ip": "192.168.0.117",  
                "port": 3260  
              }  
            ],  
            "targetPortalIP": {  
              "ips": [  
                {  
                  "ip": "10.0.1.10",  
                  "port": 10011  
                }  
              ],  
              "status": "Disabled"  
            },  
            "serverId": "hbblock_1"  
          },  
          {  
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.4",  
            "ips": [  
              {  
                "ip": "192.168.0.102",  
                "port": 3260  
              }  
            ],  
            "serverId": "hbblock_2"  
          }  
        ],  
        "serverId": "hbblock_1"  
      }  
    ]  
  }  
],  
"serverId": "hbblock_1"
```

```
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.5",
        "ips": [
            {
                "ip": "192.168.0.192",
                "port": 3260
            }
        ],
        "serverId": "hblock_3"
    }
],
"chap": {
    "name": "chap-test",
    "status": "Disabled"
},
"createTime": 1695806252000,
"serverIds": [
    "hblock_1",
    "hblock_2",
    "hblock_3"
],
"num": 3
},
{
    "targetName": "target02",
    "maxSessions": 1,
    "iSCSITargets": [
        {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.9",
            "ips": [
                {
                    "ip": "192.168.0.110",
                    "port": 3260
                }
            ],
            "serverId": "hblock_4"
```

```
    },  
    {  
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.10",  
      "ips": [  
        {  
          "ip": "192.168.0.192",  
          "port": 3260  
        }  
      ],  
      "serverId": "hblock_3"  
    }  
  ],  
  "createTime": 1695889887559,  
  "serverIds": [  
    "hblock_3",  
    "hblock_4"  
  ],  
  "num": 2  
}  
]  
}
```

## 4.4.8 查询指定 iSCSI Target

此操作用来查询指定的 iSCSI Target。

### ● 请求语法

```
GET /rest/v1/block/target/targetName HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

### ● 请求参数

参数	描述	是否必须
targetName	iSCSI Target 名称。 说明：如果不填写此项，会返回所有 Target 信息。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	否

### ● 响应结果

名称	描述
targetName	iSCSI Target 名称。
maxSessions	iSCSI Target 下每个 IQN 允许建立的最大会话数。
iSCSITargets	iSCSI Target 属性集合。包含：iqn、ips、targetPortalIP、serverId。 类型：数组
iqn	iSCSI Target IQN。
ips	iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组

ip	iSCSI Target 的 IP。
port	iSCSI Target 的端口号。
targetPortalIP.ips	iSCSI 目标门户 IP 属性集合，包含：ip、port。 类型：数组
ip	iSCSI 目标门户 IP。 说明：如果未配置，不返回该项。
port	iSCSI 目标门户端口号。 说明：如果未配置，不返回该项。
targetPortalIP.status	iSCSI 目标门户 IP 状态： <ul style="list-style-type: none"> <li>● Enabled：启用状态。</li> <li>● Disabled：停用状态。</li> </ul> 说明：如果未配置，不返回该项。
serverId	iSCSI Target 的 IQN 所在的服务器 ID。
luns	关联卷的属性，包含：lunName、number。 类型：数组
lunName	关联卷的名称。
number	卷编号。
chap.name	CHAP 认证名称。
chap.status	CHAP 认证状态： <ul style="list-style-type: none"> <li>● Enabled：启用 CHAP 认证。</li> <li>● Disabled：禁用 CHAP 认证。</li> </ul>
createTime	创建 iSCSI Target 的时间，unix 时间戳（UTC），精确到毫秒。
serverIds	iSCSI Target 对应的服务器 ID（仅集群版支持）。
num	Target 所在的服务器数量（仅集群版支持）。

● 请求示例 1

单机版：查询 iSCSI Target 名称为 `targeta` 的信息。

```
GET /rest/v1/block/target/targeta HTTP/1.1
Date: Thu, 28 Sep 2023 09:17:10 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
Connection: keep-alive
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 8f4d4a6648aa49daa3e9a3b2e6793678
Connection: keep-alive
Content-Length: 332
Date: Thu, 28 Sep 2023 09:17:10 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "targetName": "targeta",
    "maxSessions": 1,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:targeta.2",
        "ips": [
          {
            "ip": "192.168.0.32",
            "port": 3260
          }
        ],
        "targetPortalIP": {
          "ips": [
            {
              "ip": "10.1.1.10",
              "port": 20430
            }
          ]
        }
      }
    ]
  }
}
```

```
        }
      ],
      "status": "Disabled"
    },
    "serverId": "hblock_1"
  }
],
"luns": [
  {
    "lunName": "luna1",
    "number": 0
  }
],
"createTime": 1695891789350,
"num": 1
}
}
```

### ● 请求示例 2

集群版：查询 iSCSI Target 名称为 **target01** 的信息。

```
GET /rest/v1/block/target/target01 HTTP/1.1
Date: Thu, 28 Sep 2023 09:17:39 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.117:1443
Connection: keep-alive
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 5d8358567a474c0cadf4be1b808ce684
Connection: keep-alive
Content-Length: 692
Date: Thu, 28 Sep 2023 09:17:39 GMT
Content-Type: application/json;charset=utf-8
```

Server: HBlock

```
{
  "data": {
    "targetName": "target01",
    "maxSessions": 10,
    "iSCSITargets": [
      {
        "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.3",
        "ips": [
          {
            "ip": "192.168.0.117",
            "port": 3260
          }
        ],
        "targetPortalIP": {
          "ips": [
            {
              "ip": "10.0.1.10",
              "port": 10011
            }
          ]
        },
        "status": "Disabled"
      },
      {
        "serverId": "hblock_1"
      }
    ],
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.4",
      "ips": [
        {
          "ip": "192.168.0.102",
          "port": 3260
        }
      ],
      "serverId": "hblock_2"
    }
  ]
}
```

```
    },
    {
      "iqn": "iqn.2012-08.cn.ctyunapi.oos:target01.5",
      "ips": [
        {
          "ip": "192.168.0.192",
          "port": 3260
        }
      ],
      "serverId": "hblock_3"
    }
  ],
  "luns": [
    {
      "lunName": "lun01a",
      "number": 0
    },
    {
      "lunName": "lun01b",
      "number": 1
    }
  ],
  "chap": {
    "name": "chap-test",
    "status": "Disabled"
  },
  "createTime": 1695806252000,
  "serverIds": [
    "hblock_1",
    "hblock_2",
    "hblock_3"
  ],
  "num": 3
}
}
```

## 4.4.9 查询 iSCSI Target 连接

此操作用来查询 iSCSI Target 的连接信息。

### ● 请求语法

```
GET /rest/v1/block/connection/?filter=filter&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为： targetName: iSCSI Target 名称。	否
range	查询 iSCSI Target 连接的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照 iSCSI Target 名称进行排序，返回第 i 到第 j 的 iSCSI Target 连接信息。 如果输入的查询范围超过 iSCSI Target 连接的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的 iSCSI Target 连接。	否

### ● 响应结果

名称	描述
connections	连接信息集合，包含：targetName、targetConnections。 类型：数组
targetName	iSCSI Target 名称。
targetConnections	iSCSI Target 连接信息集合，包含：sessionId、iqn、clientIP、clientPort、initiatorName、targetIP。

	类型：数组
sessionId	会话 ID。
iqn	IQN。
clientIP	客户端的 IP。
clientPort	客户端的端口号。
initiatorName	连接所属的 initiator 名称。
targetIP	服务器端 IP。

### ● 请求示例

查询 iSCSI Target 名称含 **target** 的连接。

```
GET /rest/v1/block/connection/?filter=targetName:target HTTP/1.1
Date: Wed, 16 Mar 2022 02:52:21 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 596a0b14ab0b482b9ee88d28c701b90d
Connection: keep-alive
Content-Length: 457
Date: Wed, 16 Mar 2022 02:52:21 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "connections": [
      {
        "targetName": "target02",
        "targetConnections": [
```

```
{
  {
    "sessionId": "0x4",
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.7",
    "clientIP": "192.168.0.46",
    "clientPort": 51978,
    "initiatorName": "iqn.1991-05.com.microsoft:ecs-28f3",
    "targetIP": "192.168.0.121"
  },
  {
    "sessionId": "0x5",
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target02.8",
    "clientIP": "192.168.0.46",
    "clientPort": 51980,
    "initiatorName": "iqn.1991-05.com.microsoft:ecs-28f3",
    "targetIP": "192.168.0.72"
  }
]
}
]
```

## 4.4.10 删除 iSCSI Target 连接

此操作用来删除 iSCSI Target 连接。

### ● 请求语法

```
DELETE
/rest/v1/block/connection/targetName?initiatorName=initiatorName&targetIP=targetIP
HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
targetName	删除的连接所属的 iSCSI Target 名称。 类型：字符串 取值：长度范围是 1~16，可以由小写字母、数字、句点 (.) 和短横线 (-) 组成，且仅支持以字母或数字开头。	是
initiatorName	删除的连接所属的 initiator 名称。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● 若使用 Windows iSCSI 发起程序，initiatorName 为“发起程序名称”。</li> <li>● 若使用 Linux iSCSI 发起程序，initiatorName 为使用 <b>cat /etc/iscsi/initiatorname.iscsi</b> 指令取得“InitiatorName”。</li> </ul>	否
targetIP	要删除连接所对应的 iSCSI Target IP。	否

### ● 请求示例

删除 iSCSI Target 名称为 `target02`、发起程序名称为 `iqn.1991-05.com.microsoft:ecs-28f3`、targetIP 为 `192.168.0.72` 的连接。

```
DELETE /rest/v1/block/connection/target02?initiatorName=iqn.1991-05.com.microsoft:ecs-28f3&targetIP=192.168.0.72 HTTP/1.1
Date: Wed, 16 Mar 2022 03:24:59 GMT
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature
```

- 响应示例

```
HTTP/1.1 204 No Content
Date: Wed, 16 Mar 2022 03:24:59 GMT
Connection: keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: acec0eaf2909485992e90032ce587604
Server: HBlock
```

#### 4.4.11 错误码列表

HTTP status	错误码	错误信息	描述
400	InsufficientServers	This method is not allowed because the number of available servers in the cluster is not enough.	集群中可用服务器数量不足，无法操作，请解决后重试。
400	InvalidCHAPName	Value <i>value</i> at 'chap name' failed to satisfy constraint: Argument can contain letters, digits, dots(.), colons(:), underscores(_) or hyphens (-),begin with a letter or digit, and the length is between 3 and 64.	CHAP 名字不合法。
400	InvalidCHAPPassword	Value <i>value</i> at 'password' failed to satisfy constraint: Argument can contain letters, digits, or underscores(_) and the length is between 12 and 16.	密码不合法。
400	InvalidString	Value " at ' <i>argument</i> ' failed to satisfy constraint: Argument must not be an empty string.	参数不能是空字符串。
400	InvalidTargetDest	Invalid Target destination server <i>server_ID</i> .	目标服务器目前无法创建该 Target，请重新选择。
400	InvalidTargetName	Value <i>value</i> at 'target name'	Target 名称不正

		failed to satisfy constraint: Argument must contain only lowercase letters, digits, dots(.), or hyphens (-), begin with a lowercase letter or digit, and does not exceed 16 characters.	确。
400	InvalidTargetSource	Invalid Target source server <i>server_ID</i> .	Target 的源服务器目前不能进行迁移操作，请稍后再试。
400	LessThanMinValue	Value <i>value</i> at 'number' failed to satisfy constraint: Argument must be greater than or equal to <i>value</i> .	该参数必须大于或等于另一个参数。
400	MissingCHAPName	Value null at 'chap name' failed to satisfy constraint: Argument must not be null when CHAP is enabled.	启用 CHAP 时，缺少 CHAP 名称。
400	MissingCHAPPassword	Value null at 'password' failed to satisfy constraint: Argument must not be null when CHAP is enabled.	启用 CHAP 时，缺少密码。
400	MissingTargetSourceOrDest	Target source and destination server ID must exist when migrating Target server.	迁移 Target 时，源和目的服务器 ID 必须存在。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	TooManyServers	The number of servers cannot	Target 对应的服务

		exceed <i>value</i> .	器数量不能超过 <i>value</i> 个。
400	TooManyTargets	The number of Target IQNs cannot exceed 32766.	Target IQN 数量超过上限。
404	NoSuchCHAP	The Target with name <i>name</i> has no CHAP.	Target 下没有 CHAP。
404	NoSuchConnection	The connection does not exist.	连接不存在。
404	NoSuchTarget	The Target with name <i>targetName</i> does not exist.	Target 不存在。
404	NoSuchTargetIP	The Target with IP <i>targetIP</i> does not exist.	IP 为该值的 Target 不存在。
409	ConflictWithTargetMigrating	Failed to action because the Target <i>targetName</i> is migrating.	Target 迁移中，无法执行操作。
409	ConnectionExists	There are connections on this Target. Please disconnect the connections and try again.	Target 下有连接信息，请断开连接后重试。
409	InvalidServerStatus	The server status is ' <i>status</i> ', the request is invalid.	服务器当前的状态不正确，请求无效。
409	InvalidStorStatus	The HBlock status is ' <i>status</i> ', the request is invalid.	HBlock 当前的状态不正确，请求无效。
409	LUNExists	There are LUNs in this Target. Please delete all LUNs and try again.	Target 有关联的卷。
409	TargetAlreadyExists	The Target with name <i>targetName</i> already exists.	Target 已经存在

## 4.5 服务器管理

### 4.5.1 添加服务器（集群版适用）

此操作用来向集群中添加新的服务器。

请先在待添加的服务器上执行下列步骤：

- 1) 拷贝安装包至安装路径。
- 2) 解压安装包。
- 3) 执行 `./stor install` 命令安装。

然后在集群中的任意一台服务器上，执行添加服务器的操作。

注意：

- 请确保 Linux 用户具有所需要端口的权限。Linux 系统默认小于 1024 的端口不对没有 root 权限的 Linux 普通用户开放。
- 设置端口范围（portRange）时，请避免和 Linux 系统的本地临时端口（ip\_local\_port\_range）范围重合，否则可能会导致 HBlock 服务所用的端口被占用。使用命令行 `cat /proc/sys/net/ipv4/ip_local_port_range` 可以查看本地临时端口范围。

- 请求语法

```
POST /rest/v1/system/server HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "ip": ip,
  "apiPort": apiPort,
  "iSCSIPort": iSCSIPort,
  "portRange": port1-port2,
```

```

"diskPaths": [
  { "path": path,
    "capacityQuota": capacityvalue
  },
  { "path": path,
    "capacityQuota": capacityvalue
  },
  .....
  ...],
"ports":{
  "dataPort1": dataPort1,
  "managementPort1": managementPort1,
  "managementPort2": managementPort2,
  "managementPort3": managementPort3,
  "managementPort4": managementPort4,
  "managementPort5": managementPort5,
  "managementPort6": managementPort6
}
}
    
```

### ● 请求参数

参数	描述	是否必须
ip	要添加的服务器 IP。	是
apiPort	管理 API 端口。 类型：整型 取值： [1, 65535]，默认值为 1443。	否
iSCSIPort	iSCSI 端口。 类型：整型 取值： [1, 65535]，默认值为 3260。	否

<p>portRange</p>	<p>指定端口范围。存储服务以及未指定端口的服务将从此范围中自动取值。</p> <p>类型：整型</p> <p>取值：[1, 65535]，<i>port1</i> 为端口范围最小值，<i>port2</i> 为端口范围最大值，且 <i>port1</i>&lt;<i>port2</i>。<i>port1</i> 默认取值为 20000，<i>port2</i> 默认取值为 20500。</p> <p><b>说明：</b>建议指定的端口范围至少包含 500 个端口。</p>	<p>否</p>
<p>diskPaths</p>	<p>数据目录属性集合。包括 path、capacityQuota。</p> <p>类型：数组。</p>	<p>是</p>
<p>path</p>	<p>指定数据目录，支持多个数据目录，至少指定一个数据目录。数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。</p> <p><b>注意：</b>数据目录中不能有逗号（,）。</p> <p>类型：字符串列表</p>	<p>是</p>
<p>capacityQuota</p>	<p>指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。</p> <p>类型：长整型。</p> <p>取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。</p> <p><b>注意：</b>如果相同的数据目录出现多次，以第一次出现的数据目录的容量配额为准。</p>	<p>否</p>
<p>ports.dataPort1</p>	<p>数据端口 1。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	<p>否</p>

ports.managementPort1	管理服务端口 1。 类型：整型 取值：[1, 65535]。	否
ports.managementPort2	管理服务端口 2。 类型：整型 取值：[1, 65535]。	否
ports.managementPort3	管理服务端口 3。 类型：整型 取值：[1, 65535]。	否
ports.managementPort4	管理服务端口 4。 类型：整型 取值：[1, 65535]。	否
ports.managementPort5	管理服务端口 5。 类型：整型 取值：[1, 65535]。	否
ports.managementPort6	管理服务端口 6。 类型：整型 取值：[1, 65535]。	否

- 请求示例

添加 IP 为 192.168.0.202 的服务器至集群，数据目录为 /mnt/storage01 和 /mnt/storage02。

```
POST /rest/v1/system/server HTTP/1.1
Date: Tue, 23 Jan 2024 08:16:17 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 180
Host: 192.168.0.110:1443
```

```
{
  "ip": "192.168.0.202",
  "diskPaths": [
    {
      "path": "/mnt/storage01"
    },
    {
      "path": "/mnt/storage02"
    }
  ]
}
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 2707c6619d4c4eb9897ec7d826d8c901
Connection: keep-alive
Date: Tue, 23 Jan 2024 08:16:17 GMT
Content-Length: 0
Server: HBlock
```

## 4.5.2 修改服务器属性

此操作用来修改服务器属性。

**注意：** 修改端口范围（portRange）时，请避免和 Linux 系统的本地临时端口

（ip\_local\_port\_range）范围重合，否则可能会导致 HBlock 服务所用的端口被占用。使用命令行 `cat /proc/sys/net/ipv4/ip_local_port_range` 可以查看本地临时端口范围。

### ● 请求语法

```
PUT /rest/v1/system/server/serverId HTTP/1.1
Date:date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization:authorization

{
  "targetPortalIP": {
    "ips":[
      {
        "ip": ip,
        "port": port,
      }
    ],
    "status": status
  },
  "defaultPath": defaultPath,
  "portRange": port1-port2
}
```

### ● 请求参数

参数	描述	是否必须
serverId	要修改的服务器 ID。	是

targetPortalIP.ips	<p>iSCSI 目标门户 IP 属性集合，包含：<b>ip</b>、<b>port</b>。</p> <p>类型：数组。</p> <p><b>说明：</b>若服务器与客户端不在同一网段（如服务器位于内网，客户端位于外网），通过 NAT 设备（如路由器）进行连接，则需要将 NAT 设备的外网地址和端口添加到服务器，从而使得外网的客户端可以正常与该服务器的 Target 建立 iSCSI 连接。</p>	否
ip	<p>iSCSI 目标门户 IP。</p> <p><b>说明：</b>ip 和 port 必须同时存在，或者同时不存在。</p>	否
port	<p>iSCSI 目标门户端口号。</p> <p><b>说明：</b>ip 和 port 必须同时存在，或者同时不存在。</p> <p>类型：整型</p> <p>取值：[1, 65535]。</p>	否
targetPortalIP.status	<p>是否启用 iSCSI 目标门户 IP。</p> <p>类型：枚举</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 启用。</li> <li>● <b>Disabled:</b> 禁用。</li> </ul>	是
defaultPath	<p>设置默认的数据目录（仅单机版本支持）。</p> <p>数据目录必须是已经添加到 HBlock 系统中，并且状态为 <b>Normal</b> 的数据目录。</p>	否
portRange	<p>指定端口范围，用于数据服务。</p> <p>取值：整型，取值范围为[1, 65535]，<b>port1</b> 为端口范围最小值，<b>port2</b> 为端口范围最大值，且 <b>port1</b>&lt;<b>port2</b>。</p> <p><b>说明：</b>建议指定的端口范围至少包含 500 个端口。</p>	否

- 请求示例 1

单机版：修改服务器 hblock\_1 的 iSCSI 目标门户 IP 和端口修改为：10.1.0.1 和 101，设置默认数据目录为：/mnt/storage01。

```
PUT /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Fri, 29 Jul 2022 09:12:25 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 270
Host: 192.168.0.69:1443

{
  "targetPortalIP": {
    "ips": [
      {
        "ip": "10.1.0.1",
        "port": "101"
      }
    ],
    "status": "Enabled"
  },
  "defaultPath": "/mnt/storage01"
}
```

- 响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: facb844a44034035bd9254c7d901cfac
Connection: keep-alive
Date: Fri, 29 Jul 2022 09:12:25 GMT
Server: HBlock
```

- 请求示例 2

集群版：修改服务器 hblock\_1 的 iSCSI 目标门户 IP 和端口修改为：10.1.0.1 和 101。

```
PUT /rest/v1/system/server/hblock_1 HTTP/1.1
```

```
Date: Fri, 29 Jul 2022 09:00:33 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 221
Host: 192.168.0.121:1443

{
  "targetPortalIP": {
    "ips":[
      {
        "ip":"10.1.0.1",
        "port":"101"
      }
    ],
    "status":"Enabled"
  }
}
```

- 响应示例 2

```
HTTP/1.1 204 No Content
x-hblock-request-id: cbe33ebab9ad46df9fe691b71585e230
Connection: keep-alive
Date: Fri, 29 Jul 2022 09:00:33 GMT
Server: HBlock
```

- 请求示例 3

修改服务器 hblock\_1 的端口范围为 19500-20500。

```
PUT /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Tue, 26 Sep 2023 06:45:38 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 39
Host: 192.168.0.121:1443
```

```
{  
  "portRange": "19500-20500"  
}
```

- 响应示例 3

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 2271ef7c175c44cca2e197c12f2cb742  
Connection: keep-alive  
Date: Tue, 26 Sep 2023 06:45:38 GMT  
Server: HBlock
```

### 4.5.3 删除服务器属性

此操作用来删除服务器 targetPortalIP 配置。

- 请求语法

```
DELETE /rest/v1/system/server/serverId/targetportalIP HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

- 请求参数

参数	描述	是否必须
serverId	要删除 targetPortalIP 配置的服务器 ID。	是

- 请求示例

删除 hblock\_1 的 iSCSI 目标门户 IP。

```
DELETE /rest/v1/system/server/hblock_1/targetportalIP HTTP/1.1
Date: Thu, 17 Mar 2022 01:01:08 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 821c5921163f47d292aa281887c551f3
Connection: keep-alive
Date: Thu, 17 Mar 2022 01:01:09 GMT
Server: HBlock
```

## 4.5.4 移除服务器（集群版适用）

此操作用来移除服务器。

注意：

- 如果移除服务器上有 Target，该 Target 对应卷的高可用类型是 ActiveStandby，移除服务器时，业务不会中断，此卷对应的 Target 会切换到其他服务器上，客户端需要重新连接 Target 对应的新服务器 IP。
- 如果移除服务器上有 Target，该 Target 对应卷的高可用类型是 Disabled，移除服务器时，业务会中断，此卷对应的 Target 会切换到其他服务器上，客户端需要重新连接 Target 对应的新服务器 IP。但服务器移除时，会有数据丢失风险。
- 如果执行日志采集后，产生的日志保存在服务器安装目录下，在服务器移除之后，该日志将被删除。如果产生的日志保存在 HBlock 的数据目录内，并且 deleteLocalData 为 true，该日志也将被删除。
- 有服务器正在移除时，不能再移除其他服务器。如果必须移除，请使用强制移除，但有丢数据风险。

### ● 请求语法

```
DELETE /rest/v1/system/server/serverId?force=force&deleteLocalData=deleteLocalData
HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
serverId	要移除服务器的服务器 ID。	是
force	是否强制移除服务器。 <b>注意：</b> 强制移除服务器，可能造成数据丢失。 类型：布尔	否

	取值： <ul style="list-style-type: none"> <li>● <b>true</b>：是。</li> <li>● <b>false</b>：否。</li> </ul> 默认值为 <b>false</b> 。	
<b>deleteLocalData</b>	移除服务器时，是否删除服务器上 HBlock 数据目录中的数据。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：删除服务器上 HBlock 数据目录中的数据。</li> <li>● <b>false</b>：不删除服务器上 HBlock 数据目录中的数据。</li> </ul> 默认值为 <b>false</b> 。	否

● 响应结果

名称	描述
<b>luns</b>	被移除服务器上卷的属性信息集合，包含： <b>lunName</b> 、 <b>iSCSITargets</b> 。 类型：数组
<b>lunName</b>	被移除服务器上的卷名称。
<b>iSCSITargets</b>	被移除服务器上卷对应的 iSCSI Target 属性集合。包含： <b>iqn</b> 、 <b>status</b> 、 <b>ips</b> 。 类型：数组
<b>iqn</b>	被移除服务器上的卷对应的新 Target IQN。
<b>status</b>	被移除服务器上的卷对应的 Target 的状态： <ul style="list-style-type: none"> <li>● <b>Active</b>：主 Target。</li> <li>● <b>Standby</b>：热备 Target。</li> <li>● <b>Offline</b>：离线。</li> <li>● <b>Unavailable</b>：冷备 Target。</li> </ul>

ips	被移除服务器上卷对应 iSCSI Target 的 IP 属性集合，包含：ip、port。 类型：数组
ip	被移除服务器上的卷对应的新 iSCSI Target 的 IP。
port	被移除服务器上的卷对应的新的 iSCSI Target 的端口号。

### ● 请求示例

移除服务器 hblock\_4。

```
DELETE / rest/v1/system/server/hblock_4?force=false&deleteLocalData=false HTTP/1.1
Date: Thu, 17 Mar 2022 02:16:39 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 202 Accepted
x-hblock-request-id: a1546d76c10a4758ae4b19f7fe140a02
Connection: keep-alive
Content-Length: 276
Date: Thu, 17 Mar 2022 02:16:42 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "luns": [
      {
        "lunName": "lun4",
        "iSCSITargets": [
          {
            "iqn": "iqn.2012-08.cn.ctyunapi.oos:target4.15",
            "status": "Offline",
            "ips": [
              {
```

```
        "ip": "192.168.0.72",
        "port": 3260
      }
    ]
  },
  {
    "iqn": "iqn.2012-08.cn.ctyunapi.oos:target4.16",
    "status": "Standby",
    "ips": [
      {
        "ip": "192.168.0.121",
        "port": 3260
      }
    ]
  }
]
}
}
```

## 4.5.5 重启服务器上的 HBlock

此操作用来重启服务器上的 HBlock。

### ● 请求语法

```
PUT /rest/v1/system/server/serverId/restart HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "force": force
}
```

### ● 请求参数

参数	描述	是否必须
serverId	要重启的服务器 ID。	是
force	是否强制重启。 <b>注意：</b> 强制重启服务器，可能造成数据丢失。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● true：强制重启。</li> <li>● false：不强制重启。</li> </ul> 默认值为 false。	否

### ● 请求示例

重启服务器 hblock\_4 上的 HBlock 服务。

```
PUT /rest/v1/system/server/hblock_4/restart HTTP/1.1
```

```
Date: Thu, 17 Mar 2022 02:00:12 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 202 Accepted  
x-hblock-request-id: c221ede9cf764ce39a06c3e8ea930fdd  
Connection: keep-alive  
Date: Thu, 17 Mar 2022 02:00:16 GMT  
Content-Length: 0  
Server: HBlock
```

## 4.5.6 批量查询服务器

此操作用来批量查询服务器。

### ● 请求语法

```
GET /rest/v1/system/server?filter=filter&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为： <ul style="list-style-type: none"> <li>● <b>serverName</b>: 服务器名称。</li> <li>● <b>status</b>: 服务器状态（<b>Connected</b>: 已连接; <b>Disconnected</b>: 未连接; <b>Removing</b>: 移除中）。</li> </ul>	否
range	查询服务器的范围，格式为 <i>i-j</i> ， <i>i</i> 和 <i>j</i> 为正整数， $0 < j-i < 1000$ 。按照服务器 ID 进行排序，返回第 <i>i</i> 到第 <i>j</i> 的服务器。 如果输入的查询范围超过服务器的总个数，返回的结果为空，如果未指定 <b>range</b> ，则返回所有符合条件的服务器。	否

### ● 响应结果

名称	描述
servers	服务器 HBlock 属性集合，包含： <b>serverName</b> 、 <b>serverId</b> 、 <b>status</b> 、 <b>version</b> 、 <b>isMasterServer</b> 、 <b>isBaseServer</b> 、

	publicAddress、clusterAddress、 allDiskPaths、defaultPath、 recentStartTime、ports、portRange、 iSCSIPort。 类型：数组
serverName	服务器名称。
serverId	服务器 ID。
status	服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected</b>: 已连接。</li> <li>● <b>Disconnected</b>: 未连接。</li> <li>● <b>Removing</b>: 移除中。</li> </ul>
version	服务器上的 HBlock 版本。
isMasterServer	该服务器是否为 Master 节点： <ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul>
isBaseServer	该服务器是否为基础节点： <ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul>
publicAddress.ip	业务网络的 IP。
publicAddress.port	业务网络端口号。
clusterAddress.ip	集群网络的 IP。
clusterAddress.port	集群网络端口号。
allDiskPaths.usedCapacity	当前服务器所有 HBlock 数据目录对应分区的已用容量，单位为字节。
allDiskPaths.totalCapacity	当前服务器所有 HBlock 数据目录对应分区的总容量，单位是字节。
defaultPath	默认的数据目录（仅单机版支持）。

recentStartTime	HBlock 服务在该节点上最近一次成功启动的时间。 服务器未连接时，不返回此项。
ports	HBlock 端口的集合。数组，包括 portName、port。
portName	HBlock 端口名称。
port	HBlock 在当前服务器上使用的端口号。
portRange	端口范围。
iSCSIPort	iSCSI 端口。

- 请求示例 1

单机版：查询服务器。

```
GET /rest/v1/system/server HTTP/1.1
Date: Thu, 21 Dec 2023 01:58:17 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
Date: Thu, 21 Dec 2023 01:58:17 GMT
x-hblock-request-id: 0902e4e42dcc468999b573bd3c5d0dd7
Connection: keep-alive
Content-Length: 577
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "servers": [
      {
        "serverName": "ecs-9689-0915127",
```

```
"serverId": "hblock_1",
"status": "Connected",
"version": "3.5.0",
"publicAddress": {
  "ip": "192.168.0.32",
  "port": 3260
},
"clusterAddress": {
  "ip": "192.168.0.32",
  "port": 20004
},
"allDiskPaths": {
  "usedCapacity": 224587776,
  "totalCapacity": 100167352320
},
"defaultPath": "/mnt/storage01",
"recentStartTime": 1702893605991,
"ports": [
  {
    "portName": "apiPort",
    "port": 1443
  },
  {
    "portName": "webPort",
    "port": 442
  },
  {
    "portName": "managementPort1",
    "port": 20004
  },
  {
    "portName": "managementPort2",
    "port": 20003
  },
  {
```

```
        "portName": "managementPort3",
        "port": 20001
    },
    {
        "portName": "managementPort4",
        "port": 20000
    },
    {
        "portName": "managementPort6",
        "port": 20002
    }
],
"portRange": "20000-20500",
"iSCSIPort": 3260
}
]
}
```

### ● 请求示例 2

集群版：查询第 2 和第 3 个服务器。

```
GET /rest/v1/system/server?range=2-3 HTTP/1.1
Date: Thu, 21 Dec 2023 01:33:20 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
Date: Thu, 21 Dec 2023 01:33:20 GMT
x-hblock-request-id: 3db91ebb72a7452fa1251c37c896478b
Connection: keep-alive
Content-Length: 2292
```

```
Content-Type: application/json;charset=utf-8
```

```
Server: HBlock
```

```
{
  "data": {
    "servers": [
      {
        "serverName": "pm-006",
        "serverId": "hblock_2",
        "status": "Connected",
        "version": "3.5.0",
        "isMasterServer": false,
        "isBaseServer": true,
        "publicAddress": {
          "ip": "192.168.0.192",
          "port": 3260
        },
        "clusterAddress": {
          "ip": "192.168.0.192",
          "port": 20007
        },
        "allDiskPaths": {
          "usedCapacity": 5228392448,
          "totalCapacity": 100169342976
        },
        "recentStartTime": 1702628094509,
        "ports": [
          {
            "portName": "apiPort",
            "port": 1443
          },
          {
            "portName": "webPort",
            "port": 442
          }
        ]
      }
    ]
  }
}
```

```
{
  "portName": "managementPort1",
  "port": 20007
},
{
  "portName": "managementPort2",
  "port": 20011
},
{
  "portName": "managementPort3",
  "port": 20013
},
{
  "portName": "managementPort4",
  "port": 20006
},
{
  "portName": "managementPort5",
  "port": 20004
},
{
  "portName": "managementPort6",
  "port": 20008
},
{
  "portName": "dataPort1",
  "port": 20003
},
{
  "portName": "metadataPort1",
  "port": 20012
},
{
  "portName": "metadataPort2",
  "port": 20009
}
```

```
    },  
    {  
      "portName": "metadataPort3",  
      "port": 20014  
    },  
    {  
      "portName": "metadataPort4",  
      "port": 20005  
    },  
    {  
      "portName": "metadataPort5",  
      "port": 20010  
    },  
    {  
      "portName": "metadataPort6",  
      "port": 20000  
    },  
    {  
      "portName": "metadataPort7",  
      "port": 20002  
    },  
    {  
      "portName": "metadataPort8",  
      "port": 20001  
    }  
  ],  
  "portRange": "20000-20500",  
  "iSCSIPort": 3260,  
  "storagePorts": [  
    {  
      "port1": 20015,  
      "port2": 20017,  
      "port3": 20016,  
      "diskpath": "/mnt/stor",  
      "pathId": "1"  
    }  
  ]  
}
```

```
    }
  ]
},
{
  "serverName": "ecs-9689-0915140",
  "serverId": "hblock_3",
  "status": "Connected",
  "version": "3.5.0",
  "isMasterServer": false,
  "isBaseServer": true,
  "publicAddress": {
    "ip": "192.168.0.102",
    "port": 3260
  },
  "clusterAddress": {
    "ip": "192.168.0.102",
    "port": 20007
  },
  "allDiskPaths": {
    "usedCapacity": 7859613696,
    "totalCapacity": 100169342976
  },
  "recentStartTime": 1702364604218,
  "ports": [
    {
      "portName": "apiPort",
      "port": 1443
    },
    {
      "portName": "webPort",
      "port": 442
    },
    {
      "portName": "managementPort1",
      "port": 20007
    }
  ]
}
```

```
    },  
    {  
      "portName": "managementPort2",  
      "port": 20011  
    },  
    {  
      "portName": "managementPort3",  
      "port": 20013  
    },  
    {  
      "portName": "managementPort4",  
      "port": 20006  
    },  
    {  
      "portName": "managementPort5",  
      "port": 20004  
    },  
    {  
      "portName": "managementPort6",  
      "port": 20008  
    },  
    {  
      "portName": "dataPort1",  
      "port": 20003  
    },  
    {  
      "portName": "metadataPort4",  
      "port": 20005  
    },  
    {  
      "portName": "metadataPort5",  
      "port": 20010  
    },  
    {  
      "portName": "metadataPort6",
```

```
        "port": 20000
      },
      {
        "portName": "metadataPort7",
        "port": 20002
      },
      {
        "portName": "metadataPort8",
        "port": 20001
      }
    ],
    "portRange": "20000-20500",
    "iSCSIPort": 3260,
    "storagePorts": [
      {
        "port1": 20009,
        "port2": 20014,
        "port3": 20012,
        "diskpath": "/mnt/stor",
        "pathId": "1"
      }
    ]
  }
}
```

## 4.5.7 查询指定服务器

此操作用来查询指定服务器。

### ● 请求语法

```
GET /rest/v1/system/server/serverId HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

### ● 请求参数

参数	描述	是否必须
serverId	要查询的服务器 ID。 如果未指定服务器 ID，则查询所有服务器。	否

### ● 响应结果

名称	描述
serverName	服务器名称。
serverId	服务器 ID。
status	服务器状态： <ul style="list-style-type: none"> <li>● <b>Connected</b>: 已连接。</li> <li>● <b>Disconnected</b>: 未连接。</li> <li>● <b>Removing</b>: 移除中。</li> </ul>
version	服务器上的 HBlock 版本。
isMasterServer	该服务器是否为 Master 节点（仅集群版支持）： <ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul>
isBaseServer	该服务器是否为基础节点（仅集群版支持）：

	<ul style="list-style-type: none"> <li>● <b>true</b>: 是。</li> <li>● <b>false</b>: 否。</li> </ul>
<code>publicAddress.ip</code>	业务网的 IP。
<code>publicAddress.port</code>	业务网端口号。
<code>clusterAddress.ip</code>	集群网 IP 地址。
<code>clusterAddress.port</code>	集群网端口号。
<code>targetPortalIP.ips</code>	iSCSI 目标门户 IP 属性集合, 包含: <code>ip</code> 、 <code>port</code> 。 类型: 数组
<code>ip</code>	iSCSI 目标门户 IP。
<code>port</code>	iSCSI 目标门户端口号。
<code>targetPortalIP.status</code>	iSCSI 目标门户 IP 状态: <ul style="list-style-type: none"> <li>● <b>Enabled</b>: 启用状态。</li> <li>● <b>Disabled</b>: 停用状态。</li> </ul>
<code>diskPaths</code>	HBlock 数据目录信息集合, 包含: <code>path</code> 、 <code>totalCapacity</code> 、 <code>usedCapacity</code> 、 <code>usedCapacityQuota</code> 、 <code>capacityQuota</code> 、 <code>managementStatus</code> 、 <code>healthStatus</code> 、 <code>healthDetail</code> 。 类型: 数组
<code>path</code>	HBlock 数据目录。
<code>totalCapacity</code>	当前 HBlock 数据目录对应分区的总容量, 单位是字节。
<code>usedCapacity</code>	服务器上 HBlock 数据目录对应分区的已用容量, 单位是字节。
<code>usedCapacityQuota</code>	HBlock 在该数据目录的已写入量, 即已用容量配额。单位是字节。
<code>capacityQuota</code>	数据目录的容量配额。单位是字节。

	0 表示禁止写入。不输出，表示没有限制。
managementStatus	数据目录管理状态： <ul style="list-style-type: none"> <li>● Added: 数据目录已添加至系统中。</li> <li>● Removing: 数据目录正在移除中。</li> </ul>
subStatus.stage	移除数据目录所处的阶段： <ul style="list-style-type: none"> <li>● Reconfiguration: 重置中。</li> <li>● CheckingData: 检查数据。在该阶段，会同时显示下列字段的信息 totalFD、healthyFD、warningFD、errorFD、safeData、reconstructionWaitData、availFDWaitData、singleCopyLUNData、corruptedData、具体告警信息（availFDWaitWarning、singleCopyLUNWarning 或者 corruptedWarning）。</li> <li>● Executing: 执行移除数据目录</li> </ul>
subStatus.totalFD	总的故障域个数。
subStatus.healthyFD	状态为 Healthy 的故障域个数。
subStatus.warningFD	状态为 Warning 的故障域个数。
subStatus.errorFD	状态为 Error 的故障域个数。
subStatus.safeData	安全数据百分比，单位是%。
subStatus.reconstructionWaitData	需要重建的数据百分比，单位是%。
subStatus.availFDWaitData	移除磁盘或者服务器时，数据重建需要的故障域个数不足，因此引发的无法重建的数据百分比。
subStatus.singleCopyLUNData	单副本数据百分比，单位是%。
subStatus.corruptedData	已经损坏的数据百分比，单位是%。
subStatus.availFDWaitWarning	告警信息。

subStatus.singleCopyLUNWarning	根据状态不同，显示的告警信息不同：
subStatus.corruptedWarning	<ul style="list-style-type: none"> <li>● <b>availFDWaitWarning</b>: 移除磁盘或者服务器时，数据重建需要的故障域个数不足。</li> <li>● <b>singleCopyLUNWarning</b>: 单副本卷数据所在磁盘或者服务器被移除。</li> <li>● <b>corruptedWarning</b>: 故障域异常导致数据损坏。</li> </ul>
healthStatus	数据目录健康状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 数据目录处于健康状态，可正常读写，且数据目录所在磁盘使用率未超过 90%。</li> <li>● <b>Warning</b>: 数据目录处于警告状态，可读，但存在以下情况的任意一种：慢盘；数据目录所在磁盘使用率超过 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> <li>● <b>Error</b>: 数据目录错误状态，无法访问，原因可能是：所在磁盘出现 I/O 错误导致无法读写，数据目录未正确挂载等。</li> </ul>
healthDetail	数据目录健康状态详情： <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此字段为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>
defaultPath	默认的数据目录（仅单机版支持）。
recentStartTime	HBlock 服务在该节点上最近一次成功启动的时间。 服务器未连接时，不返回此项。
ports	HBlock 端口的集合。数组，包括 portName、

	port。
portName	HBlock 端口名称。
port	HBlock 在当前服务器上使用的端口号。
portRange	端口范围。
iSCSIPort	iSCSI 端口。
storagePorts	存储服务端口集合（仅集群版支持）。 数组，包括 port1、port2、port3、 diskpath、pathId。
port1	存储服务端口 1。
Port2	存储服务端口 2。
port3	存储服务端口 3。
diskpath	存储服务的数据目录。
pathId	数据目录 ID。

- 请求示例 1

单机版：查询服务器 ID 为 hblock\_1 的信息。

```
GET /rest/v1/system/server/hblock_1 HTTP/1.1
Date: Thu, 21 Dec 2023 02:53:14 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: d3ba86887cdf4e22a21d8afa092b4f6a
Connection: keep-alive
Content-Length: 800
Date: Thu, 21 Dec 2023 02:53:14 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "serverName": "ecs-9689-0915127",
    "serverId": "hblock_1",
    "status": "Connected",
    "version": "3.5.0",
    "publicAddress": {
      "ip": "192.168.0.32",
      "port": 3260
    },
    "clusterAddress": {
      "ip": "192.168.0.32",
      "port": 20004
    },
    "diskPaths": [
      {
        "path": "/mnt/storage01",
        "usedCapacity": 224587776,
        "totalCapacity": 100167352320,
        "usedCapacityQuota": 94208,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
        "healthDetail": []
      }
    ],
    "defaultPath": "/mnt/storage01",
    "recentStartTime": 1702893605991,
    "ports": [
      {
        "portName": "apiPort",
        "port": 1443
      },
      {
        "portName": "webPort",
```

```

        "port": 442
    },
    {
        "portName": "managementPort1",
        "port": 20004
    },
    {
        "portName": "managementPort2",
        "port": 20003
    },
    {
        "portName": "managementPort3",
        "port": 20001
    },
    {
        "portName": "managementPort4",
        "port": 20000
    },
    {
        "portName": "managementPort6",
        "port": 20002
    }
],
"portRange": "20000-20500",
"iSCSIPort": 3260
}
}

```

- 请求示例 2

集群版：查询服务器 ID 为 hblock\_2 的信息。

```

GET /rest/v1/system/server/hblock_2 HTTP/1.1
Date: Thu, 21 Dec 2023 02:58:06 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443

```

- 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 06e98d7074e143e09a8b2fde1373b0f8
Connection: keep-alive
Content-Length: 1346
Date: Thu, 21 Dec 2023 02:58:06 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "serverName": "pm-006",
    "serverId": "hblock_2",
    "status": "Connected",
    "version": "3.5.0",
    "isMasterServer": false,
    "isBaseServer": true,
    "publicAddress": {
      "ip": "192.168.0.192",
      "port": 3260
    },
    "clusterAddress": {
      "ip": "192.168.0.192",
      "port": 20007
    },
    "diskPaths": [
      {
        "path": "/mnt/stor",
        "usedCapacity": 5228392448,
        "totalCapacity": 100169342976,
        "usedCapacityQuota": 33726464,
        "capacityQuota": 96636764160,
        "managementStatus": "Added",
        "healthStatus": "Healthy",
      }
    ]
  }
}
```

```
        "healthDetail": []
    }
],
"recentStartTime": 1702628094509,
"ports": [
    {
        "portName": "apiPort",
        "port": 1443
    },
    {
        "portName": "webPort",
        "port": 442
    },
    {
        "portName": "managementPort1",
        "port": 20007
    },
    {
        "portName": "managementPort2",
        "port": 20011
    },
    {
        "portName": "managementPort3",
        "port": 20013
    },
    {
        "portName": "managementPort4",
        "port": 20006
    },
    {
        "portName": "managementPort5",
        "port": 20004
    },
    {
        "portName": "managementPort6",
```

```
        "port": 20008
    },
    {
        "portName": "dataPort1",
        "port": 20003
    },
    {
        "portName": "metadataPort1",
        "port": 20012
    },
    {
        "portName": "metadataPort2",
        "port": 20009
    },
    {
        "portName": "metadataPort3",
        "port": 20014
    },
    {
        "portName": "metadataPort4",
        "port": 20005
    },
    {
        "portName": "metadataPort5",
        "port": 20010
    },
    {
        "portName": "metadataPort6",
        "port": 20000
    },
    {
        "portName": "metadataPort7",
        "port": 20002
    },
    {
```

```
        "portName": "metadataPort8",
        "port": 20001
    }
],
"portRange": "20000-20500",
"iSCSIPort": 3260,
"storagePorts": [
    {
        "port1": 20015,
        "port2": 20017,
        "port3": 20016,
        "diskpath": "/mnt/stor",
        "pathId": "1"
    }
]
}
```

### ● 请求示例 3

集群版：集群中有 4 台服务器，移除 hblock\_4，查询 hblock\_4 的服务器信息。

```
GET /rest/v1/system/server/hblock_4 HTTP/1.1
Date: Tue, 23 Jan 2024 08:06:07 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

### ● 响应示例 3

```
HTTP/1.1 200 OK
x-hblock-request-id: b1368719317b4f26bee52befa0b5230e
Connection: keep-alive
Content-Length: 1030
Date: Tue, 23 Jan 2024 08:06:07 GMT
Content-Type: application/json;charset=utf-8
```

Server: HBlock

```
{
  "data": {
    "serverName": "ecs-9689-0915141",
    "serverId": "hblock_4",
    "status": "Removing",
    "version": "3.5.0",
    "isMasterServer": false,
    "isBaseServer": false,
    "publicAddress": {
      "ip": "192.168.0.202",
      "port": 3260
    },
    "clusterAddress": {
      "ip": "192.168.0.202",
      "port": 20001
    },
    "diskPaths": [
      {
        "path": "/mnt/storage01",
        "usedCapacity": 1763405824,
        "totalCapacity": 100167352320,
        "usedCapacityQuota": 96944889,
        "managementStatus": "Removing",
        "healthStatus": "Warning",
        "healthDetail": [
          "StopWritingWhenRemoving"
        ]
      }
    ],
    "recentStartTime": 1705996162413,
    "ports": [
      {
        "portName": "apiPort",
```

```
        "port": 1443
    },
    {
        "portName": "webPort",
        "port": 442
    },
    {
        "portName": "managementPort1",
        "port": 20001
    },
    {
        "portName": "managementPort2",
        "port": 20003
    },
    {
        "portName": "managementPort3",
        "port": 20006
    },
    {
        "portName": "managementPort4",
        "port": 20000
    },
    {
        "portName": "managementPort5",
        "port": 20004
    },
    {
        "portName": "managementPort6",
        "port": 20002
    },
    {
        "portName": "dataPort1",
        "port": 20005
    }
],
```

```
"portRange": "20000-20500",  
"iSCSIPort": 3260,  
"storagePorts": [  
  {  
    "port1": 20007,  
    "port2": 20009,  
    "port3": 20008,  
    "diskpath": "/mnt/storage01",  
    "pathId": "1"  
  }  
]  
}
```

## 4.5.8 添加数据目录

此操作用来添加指定服务器的数据目录至 HBlock。一次可以添加多个数据目录。

**说明：**对于新增的数据目录，建议设置开机自动挂载，或使用已设置自动挂载的目录或子目录。

**注意：**每台服务器最多只能添加 100 个数据目录。

### ● 请求语法

```
POST /rest/v1/system/server/serverId/diskpaths HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "diskPaths": [
    { "path":path,
      "capacityQuota": capacityvalue
    },
    { "path":path,
      "capacityQuota": capacityvalue
    },
    .....
  ...]
}
```

### ● 请求参数

参数	描述	是否必须
serverId	要添加的数据目录所属服务器 ID。	是
diskPaths	数据目录属性集合。包括 path、capacityQuota。	是

	类型：数组。	
path	<p>指定添加的数据目录。数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。</p> <p><b>注意：</b>数据目录中不能有逗号（,）。</p> <p>类型：字符串列表</p>	是
capacityQuota	<p>指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。</p> <p>类型：长整型。</p> <p>取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认无限制写入。</p> <p><b>注意：</b>如果相同的数据目录出现多次，以第一次出现的数据目录的容量配额为准。</p>	否

### ● 请求示例

添加服务器 hblock\_2 的数据目录。

```

POST /rest/v1/system/server/hblock_2/diskpaths HTTP/1.1
Date: Wed, 27 Sep 2023 05:51:20 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 137
Host: 192.168.0.117:1443

{
  "diskPaths": [
    { "path": "/mnt/storage01",
      "capacityQuota": 1024000000
    },
    { "path": "/mnt/storage02"}
  ]
}
    
```

```
]
}
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: f0ec3fbbf3154cc99a671c3be437852a
Connection: keep-alive
Date: Wed, 27 Sep 2023 05:51:20 GMT
Content-Length: 0
Server: HBlock
```

## 4.5.9 修改数据目录的容量配额

此操作用来修改指定服务器数据目录的容量配额。

### ● 请求语法

```
PUT /rest/v1/system/server/serverId/diskpaths HTTP/1.1
```

```
Date: date
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: length
```

```
Host: ip:port
```

```
Authorization: authorization
```

```
{
  "diskPaths": [
    { "path":path,
      "capacityQuota": capacityvalue
    },
    { "path":path,
      "capacityQuota": capacityvalue
    },
    .....
  ...]
}
```

### ● 请求参数

参数	描述	是否必须
serverId	要修改的数据目录所属服务器 ID。	是
diskPaths	数据目录属性集合。包括 path、capacityQuota。 类型：数组。	是
path	指定要修改容量配额的数据目录。 类型：字符串列表	是

<b>capacityQuota</b>	指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 类型：长整型。 取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。 <b>注意：</b> 如果相同的数据目录出现多次，以第一次出现的数据目录的容量配额为准。	是
----------------------	--	---

- 请求示例

修改服务器 hblock\_2 数据目录/mnt/stor 的容量配额。

```

PUT /rest/v1/system/server/hblock_2/diskpaths HTTP/1.1
Date: Wed, 11 Oct 2023 08:06:57 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 128
Host: 192.168.0.117:1443

{
  "diskPaths": [
    {
      "path": "/mnt/stor",
      "capacityQuota": 2048000000
    }
  ]
}
    
```

- 响应示例

```

HTTP/1.1 204 No Content
x-hblock-request-id: 45f9f3b6359b48a8a663bb053d2d2731
    
```

```
Connection: keep-alive
Date: Wed, 11 Oct 2023 08:06:57 GMT
Content-Length: 0
Server: HBlock
```

## 4.5.10 移除数据目录

此操作用来移除数据目录。

注意：

- 强制移除数据目录，存在数据丢失风险。
- 对于单机版，如果要移除的数据目录为默认数据目录，需要先指定其他数据目录为默认数据目录，才可以移除该目录。
- 如果执行日志采集后，产生的日志保存在 HBlock 的数据目录内，移除该数据目录时 `deleteLocalData` 为 `true`，该日志将被删除。
- 有数据目录正在移除时，不能再移除其他数据目录。如果必须移除，请使用强制移除，但有丢数据风险。

### ● 请求语法

```
DELETE /rest/v1/system/server/serverId/diskpaths HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

```
{
  "diskPaths": [
    path,
    path
  ],
  "force": force,
  "deleteLocalData": deleteLocalData
}
```

### ● 请求参数

参数	描述	是否必须
----	----	------

serverId	要移除数据目录的服务器 ID。	是
diskPaths	要移除的 HBlock 数据目录。一次可以移除多个数据目录。	是
force	是否强制移除数据目录。 <b>注意：</b> 强制移除数据目录，可能造成数据丢失。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：强制移除数据目录。</li> <li>● <b>false</b>：不强制移除数据目录。</li> </ul> 默认值为 <b>false</b> 。	否
deleteLocalData	是否删除数据目录上的数据。 类型：布尔 取值： <ul style="list-style-type: none"> <li>● <b>true</b>：删除数据目录上的数据。</li> <li>● <b>false</b>：不删除数据目录上的数据。</li> </ul> 默认值为 <b>false</b> 。	否

### ● 请求示例

移除服务器 ID 为 hblock\_2 的数据目录/mnt/storage2。

```

DELETE /rest/v1/system/server/hblock_2/diskpaths HTTP/1.1
Date: Fri, 01 Apr 2022 01:53:09 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 124
Host: 192.168.0.121:1443

{
  "diskPaths": [
    "/mnt/storage02"
  ],
  "force": "false",

```

```
"deleteLocalData": "false"  
}
```

- 响应示例

```
HTTP/1.1 202 Accepted  
Date: Fri, 1 Apr 2022 01:53:12 GMT  
x-hblock-request-id: 5f1bfda64dc24879b7ed62d4b366f07d  
Connection: keep-alive  
Content-Length: 0  
Server: HBlock
```

### 4.5.11 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotDeleteServer	The server can not be deleted. Please stop the server, prepare a new server with same IP, and recover it.	服务器不能被移除。
400	CanNotRemoveDefaultPath	This path <i>path</i> is the default disk path, please specify a new default disk path first.	无法移除默认数据目录。
400	DeleteServerForcibly	Force delete should be used when deleting a server. When using this option, there may be a risk of data loss.	移除服务器时，必须使用强制移除，且有丢数据的风险。
400	ExceedMaxPathCount	You can only add 100 local storage paths.	每台服务器最多只能添加 100 个数据目录。
400	InsufficientPath	The disk path can not be deleted because the server must have at least one disk path.	数据目录不能被移除，因为服务器至少要有 一个数据目录。
400	InsufficientSpace	This method is not allowed because there is no more space to store data.	不能执行此操作，因为存储空间不足。
400	InvalidDiskPath	The disk paths do not exist: <i>serverIP:diskPaths</i>	因数据目录有问题导致操作失败。

		<p>[,diskPaths...] [;</p> <p><i>serverIP:diskPaths[,diskPaths..]</i>.</p> <p>The disk paths must be greater than or equal to 1 GiB:</p> <p><i>serverIP:diskPaths</i></p> <p>[,diskPaths...] [;</p> <p><i>serverIP:diskPaths[,diskPaths..]</i>.</p> <p>The disk paths can not access:</p> <p><i>serverIP:diskPaths</i></p> <p>[,diskPaths...] [;</p> <p><i>serverIP:diskPaths[,diskPaths..]</i>.</p> <p>The disk paths must be directory: <i>serverIP:diskPaths</i></p> <p>[,diskPaths...] [;</p> <p><i>serverIP:diskPaths[,diskPaths..]</i>.</p> <p>The disk paths include incompatible data:</p> <p><i>serverIP:diskPaths</i></p> <p>[,diskPaths...] [;</p> <p><i>serverIP:diskPaths[,diskPaths..]</i>.</p> <p>The disk paths can not contain commas:</p>	
--	--	--	--

		<p><i>serverIP:diskPaths</i> [,<i>diskPaths...</i>] [; <i>serverIP:diskPaths</i>[,<i>diskPaths...</i>].]</p> <p>The disk paths is already on the server and the status is removing:</p> <p><i>serverIP:diskPaths</i> [,<i>_diskPaths_...</i>]</p>	
400	InvalidDiskPathCapacityQuota	<p>The capacity quota for the following disk paths failed to satisfy constraint. It must be integer and not greater than the total capacity of disk path.</p> <p><i>serverIP:diskPaths</i> [,<i>diskPaths...</i>] [;<i>serverIP:diskPaths</i>[,<i>diskPaths...</i>].]</p>	容量配额参数错误。
400	InvalidLong	<p>Value <i>value</i> at '<i>argument</i>' failed to satisfy constraint: Argument must be of type long.</p>	参数填写错误，取值必须为长整型。
400	InvalidPathStatus	<p>The path <i>path</i> status is '<i>status</i>', the request is invalid.</p>	数据目录状态不正确，无法执行操作。
400	InvalidTargetPortalIP	<p>Value at '<i>ips</i>' failed to satisfy constraint: only support set one</p>	iSCSI 目标门户 IP、Port 只允许输入一

		Target portal IP.	组。
400	MissingTargetPortalIP	Value null at 'ip' failed to satisfy constraint: Argument must not be null when Target portal IP is enabled.	iSCSI 目标门户 IP 不能为空。
400	MissingTargetPortalIPorPort	Target portal IP and port must exist at the same time.	iSCSI 目标门户 IP 和 Port 要同时存在。
400	MissingTargetPortalPort	Value null at 'port' failed to satisfy constraint: Argument must not be null when Target portal IP is enabled.	iSCSI 目标门户 Port 不能为空。
400	PathUsed	The path <i>path</i> is being used by LUN <i>lunName</i> , <i>lunName</i> ..., please delete the LUN first.	路径正在被卷使用，请先删除卷后再重试。
400	RemoveFaultDomainRisk	Can not remove multiple fault domains at the same time. You can use force remove. When using this option, there may be a risk of data loss.	无法同时移除集群中的多个故障域。
400	RemovePathRisk	The path <i>path</i> status is ' <i>status</i> '. Removing the path may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.	数据目录状态不正常，移除数据目录可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。

400	RemovePathRisk	<p>There is LUN with redundancy mode, which has the minimum requirement for the number of paths.</p> <p>LUN: <i>lunName1</i>[, <i>lunName2</i>....]</p> <p>Removing the path may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.</p>	<p>移除数据目录可能会引起数据丢失。只能使用强制移除。请注意，强制移除，会产生数据丢失风险，请谨慎操作。</p> <p>数据目录移除后会造成卷数据损毁： <i>lunName1</i>[, <i>lunName2</i>....]。</p>
400	RemoveServerRisk	<p>There is LUN with high availability disabled, and the LUN is connected with the client.</p> <p>LUN: <i>lunName1</i>[, <i>lunName2</i>....]</p> <p>There is LUN with redundancy mode, which has the minimum requirement for the number of servers.</p> <p>LUN: <i>lunName1</i>[, <i>lunName2</i>....]</p> <p>There is Target IQN which has no available server to migrate,</p>	<p>有单控卷，且处于连接状态，移除服务器会导致卷冗余度失败。</p> <p>移除服务器会导致卷冗余度失败。</p> <p>移除服务器之后，该服务器上面部分的IQN无法在其他机器上重建。</p> <p>移除服务器时，有可能丢数据。如果必须移除，请使用强制移</p>

		<p>removing server will cause the IQN deleted.</p> <p>Target: TargetIQN1_[, TargetIQN2....]_</p> <p>Removing the server may cause data loss. You can use force remove. When using this option, there may be a risk of data loss.</p>	除。
400	RestartServerRisk	<p>There is no more space to store data or HBlock encountered an internal error. You can use force restart. When using this option, there may be a risk of data loss.</p>	重启服务器失败。可以强制重启，但有丢失数据风险。
400	StandaloneModeNotAllowed	<p>'operation' is not supported by standalone mode of HBlock.</p>	单机版本 HBlock 不支持该操作。
404	NoSuchPath	<p>The path <i>path</i> does not exist in the HBlock.</p>	路径不存在。
404	NoSuchServer	<p>The server with ID <i>serverId</i> does not exist.</p>	服务器不存在。
404	NoTargetPortalIP	<p>TargetPortalIP is not configured.</p>	没有配置 Target portal IP。
409	InvalidServerStatus	<p>The server status is '<i>status</i>', the request is invalid.</p>	服务器当前的状态不正确，请求无效。
409	PortConflict	<p>Error: the following port is/ports are in use.</p>	端口冲突。

		<i>server_ID/IP/local server: port_name port[, port_name port...]</i>	
409	RemovingAnotherPath	There is another path being removed. You can use force remove. When using this option, there may be a risk of data loss.	有数据目录正在移除时，不能再移除其他数据目录。如果必须移除，请使用强制移除，但有丢数据风险。
409	RemovingAnotherServer	There is another server being removed. You can use force remove. When using this option, there may be a risk of data loss.	有服务器正在移除时，不能再移除其他服务器。如果必须移除，请使用强制移除，但有丢数据风险。
409	ServerAlreadyExists	The server with IP <i>IP</i> already exists.	服务器 IP 已经存在。

## 4.6 查询 HBlock 信息

此操作用来查询 HBlock 信息，包括 HBlock ID、HBlock 名称、HBlock 序列号、服务器数量、故障域（集群版）、初始化时间、试用期信息及 HBlock 的管理员用户名。

### ● 请求语法

```
GET /rest/v1/system/info HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 响应结果

名称	描述
storId	HBlock ID。
storName	HBlock 的名称。
serialId	HBlock 序列号。
serverNumbers	HBlock 服务器数量。
faultDomain	故障域类型（仅集群版）： <ul style="list-style-type: none"> <li>● path: 数据目录级别的故障域。</li> <li>● server: 服务器级别的故障域。</li> </ul>
setupTime	系统初始化时间。unix 时间戳（UTC），精确到毫秒。
trialStatus	试用期标识： <ul style="list-style-type: none"> <li>● Trial: 处于试用期内，未导入软件许可证。</li> <li>● TrialExpired: 试用期已过，未导入软件许可证。</li> <li>● NonTrialVersion: 非试用版本，已导入软件许可证。</li> </ul>
trialExpireTime	试用期到期时间。unix 时间戳（UTC），精确到毫秒。
userName	HBlock 的管理员用户名。

- 请求示例 1

单机版：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Wed, 24 Jan 2024 08:00:03 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.32:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 62d8830817844ebba996c0b195bff655
Connection: keep-alive
Content-Length: 258
Date: Wed, 24 Jan 2024 08:00:03 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "storId": "B914B0E1-CF85-4954-8547-4902AE2B8B44",
    "storName": "stor",
    "serialId": "B914B0E1-CF85-4954-8547-4902AE2B8B44-0201-030500",
    "serverNumbers": 1,
    "setupTime": 1706082061964,
    "trialStatus": "Trial",
    "trialExpireTime": 1713858061964,
    "userName": "storuser"
  }
}
```

- 请求示例 2

集群版：查询 HBlock 信息。

```
GET /rest/v1/system/info HTTP/1.1
Date: Wed, 24 Jan 2024 07:39:24 GMT
```

```
Authorization: HBlock userName:signature  
Host:192.168.0.110:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: a584aa8c2b4c40d5919c7550a76b300b  
Connection: keep-alive  
Content-Length: 280  
Date: Wed, 24 Jan 2024 07:39:24 GMT  
Content-Type: application/json;charset=utf-8  
Server: HBlock  
  
{  
  "data": {  
    "storId": "22C5E31C-352D-45B5-A63A-483A8D773325",  
    "storName": "stor1",  
    "serialId": "22C5E31C-352D-45B5-A63A-483A8D773325-0202-030500",  
    "serverNumbers": 3,  
    "faultDomain": "path",  
    "setupTime": 1706079588093,  
    "trialStatus": "Trial",  
    "trialExpireTime": 1713855588093,  
    "userName": "storuser"  
  }  
}
```

## 4.7 查询 HBlock 服务状态

此操作用来查询 HBlock 状态，包括系统、故障域（集群版）、服务器、数据目录、卷的状态。

### ● 请求语法

```
GET /rest/v1/system/status HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

### ● 响应结果

名称	描述
system.status	HBlock 服务状态： <ul style="list-style-type: none"> <li>● <b>Upgrading</b>: 升级中。</li> <li>● <b>Uninstalling</b>: 卸载中。</li> <li>● <b>Working</b>: 运行中。</li> </ul>
system.licenseStatus	软件许可证状态： <ul style="list-style-type: none"> <li>● <b>Expired</b>: 软件许可证已过期。</li> <li>● <b>Effective</b>: 软件许可证已生效。</li> <li>● <b>None</b>: 还未加载软件许可证。</li> </ul>
faultDomain.level	故障域级别（仅集群版支持）： <ul style="list-style-type: none"> <li>● <b>path</b>: 数据目录级别的故障域。</li> <li>● <b>server</b>: 服务器级别的故障域。</li> </ul>
faultDomain.status.healthy	状态为健康的故障域个数（仅集群版支持）。
faultDomain.status.warning	状态为告警的故障域个数（仅集群版支持）。
faultDomain.status.error	状态为错误的故障域个数（仅集群版支持）。
server.status.connected	状态为连接的服务器个数。

server.status.disconnected	状态为断开连接的服务器个数。
server.status.removing	状态为正在移除的服务器个数。
disk.status.healthy	状态为健康的磁盘个数。
disk.status.warning	状态为告警的磁盘个数。
disk.status.error	状态为错误的磁盘个数。
disk.usage.used	HBlock 数据目录对应分区的已用容量，单位是字节。
disk.usage.total	HBlock 数据目录对应分区的总容量，单位是字节。
lun.status.number	卷的个数。
lun.status.localCapacity	卷的容量，单位是字节。
lun.data.normal	正常状态的卷数量百分比。
lun.data.lowRedundancy	低冗余状态的卷数据量百分比。
lun.data.error	错误状态的卷数据量百分比。

- 请求示例

单机版：查询 HBlock 状态。

```
GET /rest/v1/system/status HTTP/1.1
Date: Fri, 13 Oct 2023 03:08:59 GMT
Authorization: HBlock userName:signature
Host:192.168.0.202:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: e7e5867a029c40e4be956a51adb99b88
Connection: keep-alive
Content-Length: 339
Date: Fri, 13 Oct 2023 03:08:59 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
```

```
"data": {
  "system": {
    "status": "Working",
    "licenseStatus": "None"
  },
  "server": {
    "status": {
      "connected": 1,
      "disconnected": 0,
      "removing": 0
    }
  },
  "disk": {
    "status": {
      "healthy": 1,
      "warning": 0,
      "error": 0
    },
    "usage": {
      "used": 738648064,
      "total": 100167352320
    }
  },
  "lun": {
    "status": {
      "number": 1,
      "localCapacity": 32212254720
    },
    "data": {
      "normal": 100,
      "lowRedundancy": 0,
      "error": 0
    }
  }
}
```

```
}
```

- 请求示例 2

集群版：查询 HBlock 状态。

```
GET /rest/v1/system/status HTTP/1.1
Date: Fri, 13 Oct 2023 03:04:58 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

- 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 7776f775d74043ea9637e1977594ea4b
Connection: keep-alive
Content-Length: 276
Date: Fri, 13 Oct 2023 03:04:58 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "system": {
      "status": "Working",
      "licenseStatus": "Effective"
    },
    "faultDomain": {
      "level": "path",
      "status": {
        "healthy": 5,
        "warning": 0,
        "error": 0
      }
    },
    "server": {
      "status": {
```

```
        "connected": 4,  
        "disconnected": 0,  
        "removing": 0  
    }  
},  
"disk": {  
    "status": {  
        "healthy": 5,  
        "warning": 0,  
        "error": 0  
    },  
    "usage": {  
        "used": 5344075776,  
        "total": 500844724224  
    }  
},  
"lun": {  
    "status": {  
        "number": 6,  
        "localCapacity": 542239621120  
    },  
    "data": {  
        "normal": 100,  
        "lowRedundancy": 0,  
        "error": 0  
    }  
}  
}
```

## 4.8 故障域（集群版适用）

### 4.8.1 查询故障域

此操作用来查询故障域详情。

#### ● 请求语法

```
GET /rest/v1/system/faultdomain?filter=filter&range=i-j HTTP/1.1
```

Date: *date*

Content-Type: application/json; charset=utf-8

Content-Length: *length*

Host: *ip:port*

Authorization: *authorization*

#### ● 请求参数

参数	描述	是否必须
filter	<p>查询的过滤条件。可以选择多个查询，如果是或的关系，使用 <b>or</b> 将查询条件分隔开；如果是与的关系，使用 <b>and</b> 将查询条件分隔开。</p> <p>支持的过滤条件类型包括：</p> <ul style="list-style-type: none"> <li>● <b>name</b>: 故障域名称。                             <ul style="list-style-type: none"> <li>■ 服务器级别的故障域：按 <i>serverId</i> 查询。</li> <li>■ 数据目录级别的故障域：按 <i>serverId:diskpath</i> 查询。</li> </ul> </li> <li>● <b>status</b>: 故障域的健康状态                             <ul style="list-style-type: none"> <li>■ <b>Healthy</b>: 正常。</li> <li>■ <b>Warning</b>: 告警。</li> <li>■ <b>Error</b>: 错误。</li> </ul> </li> </ul>	否

range	查询故障域的范围，格式为 i-j，i 和 j 为正整数， 0<j-i<1000。按照服务器 ID 进行排序，返回第 i 到 第 j 的故障域。  如果输入的查询范围超过故障域的总个数，返回的 结果为空，如果未指定 range，则返回所有符合条 件的故障域。	否
-------	--	---

● 响应结果

名称	描述
level	故障域的级别： <ul style="list-style-type: none"> <li>● path：数据目录级别的故障域。</li> <li>● server：服务器级别的故障域。</li> </ul>
faultDomain	故障域属性集合， <ul style="list-style-type: none"> <li>● 如果故障域级别是数据目录，包含：name、tpye、status、healthDetail。</li> <li>● 如果故障域级别是服务器，包含：name、tpye、status、subNode。</li> </ul>
name	具体故障域名称。
type	故障域类型： <ul style="list-style-type: none"> <li>● path：数据目录的故障域。</li> <li>● server：服务器的故障域。</li> </ul>
status	<ul style="list-style-type: none"> <li>● 如果故障域级别是 path，则为数据服务状态的健康状态：                             <ul style="list-style-type: none"> <li>■ Healthy：健康，数据目录可正常读写，且数据目录所在磁盘使用率未超过 90%。</li> <li>■ Warning：警告，数据可读，但存在以下情况的任意一种：慢盘；数据目录所在磁盘使用率超过 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>■ <b>Error</b>: 错误，数据目录无法访问，原因可能是：所在磁盘出现 I/O 错误导致无法读写，数据目录未正确挂载等。</li> <li>● 如果故障域级别是 <b>server</b>，则为数据目录的健康状态：             <ul style="list-style-type: none"> <li>■ <b>Healthy</b>: 健康，故障域下面的所有数据服务全部都是 <b>Healthy</b> 状态。</li> <li>■ <b>Warning</b>: 警告，故障域下面的数据服务部分是 <b>Warning</b> 或 <b>Error</b> 状态。</li> <li>■ <b>Error</b>: 错误，故障域下面的所有数据服务全部都是 <b>Error</b> 状态。</li> </ul> </li> </ul>
healthDetail	数据目录的健康状态详情。 <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此列为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>
subNodes	故障域级别为服务器，子节点集合，包括 <b>name</b> 、 <b>type</b> 、 <b>status</b> 、 <b>healthDetail</b> 。
name	具体数据目录。
type	故障域的级别。
status	数据目录的健康状态： <ul style="list-style-type: none"> <li>● <b>Healthy</b>: 数据目录可正常读写，且数据目录所在磁盘使用率未超过 90%。</li> <li>● <b>Warning</b>: 数据目录可读，但存在以下情况的任意一种：慢盘；数据目录所在磁盘使用率超过 90%；磁盘剩余空间不足 1GiB；或者 HBlock 对这个目录停写。</li> <li>● <b>Error</b>: 数据目录无法访问，原因可能是：所在磁盘出现 I/O 错误导致无法读写，数据目录未正确挂载等</li> </ul>
healthDetail	数据目录的健康状态详情。 <ul style="list-style-type: none"> <li>● 如果健康状态为 <b>Healthy</b>，此列为空。</li> <li>● 如果健康状态为 <b>Warning</b> 或 <b>Error</b>，显示警告或错误的详细信息。</li> </ul>

- 请求示例 1

故障域为数据目录级别，查询数据目录包含 hblock 的故障域情况。

```
PUT /rest/v1/system/faultdomain?filter=name:hblock&range=1-3 HTTP/1.1
Date: Tue, 10 Oct 2023 08:44:00 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.192:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 1601b61c8d6744099a40576ff29fe490
Connection: keep-alive
Content-Length: 645
Date: Tue, 10 Oct 2023 08:44:00 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "level": "path",
    "faultDomain": [
      {
        "name": "hblock_1:/mnt/stor",
        "type": "path",
        "status": "Healthy",
        "healthDetail": []
      },
      {
        "name": "hblock_2:/mnt/stor",
        "type": "path",
        "status": "Healthy",
        "healthDetail": []
      },
      {
        "name": "hblock_3:/mnt/stor",
```

```
        "type": "path",
        "status": "Healthy",
        "healthDetail": []
      }
    ]
  }
}
```

### ● 请求示例 2

故障域为服务器级别，查询故障域情况。

```
PUT /rest/v1/system/faultdomain HTTP/1.1
Date: Fri, 23 Feb 2024 06:08:03 GMT
Authorization: HBlock userName:signature
Host: 192.168.0.110:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 1601b61c8d6744099a40576ff29fe490
Connection: keep-alive
Content-Length: 800
Date: Fri, 23 Feb 2024 06:08:03 GMT
Content-Type: application/json; charset=utf-8
Server: HBlock

{
  "data": {
    "level": "server",
    "faultDomain": [
      {
        "name": "hblock_1",
        "type": "server",
        "status": "Warning",
```

```
    "subNodes": [  
      {  
        "name": "/mnt/stor",  
        "type": "path",  
        "status": "Healthy",  
        "healthDetail": []  
      },  
      {  
        "name": "/mnt/storage01",  
        "type": "path",  
        "status": "Error",  
        "healthDetail": [  
          "DataServiceConnectionFailed",  
          "DataServiceExitAbnormally"  
        ]  
      }  
    ]  
  },  
  {  
    "name": "hblock_2",  
    "type": "server",  
    "status": "Healthy",  
    "subNodes": [  
      {  
        "name": "/mnt/stor",  
        "type": "path",  
        "status": "Healthy",  
        "healthDetail": []  
      }  
    ]  
  },  
  {  
    "name": "hblock_3",  
    "type": "server",  
    "status": "Healthy",
```

```
    "subNodes": [  
      {  
        "name": "/mnt/stor",  
        "type": "path",  
        "status": "Healthy",  
        "healthDetail": []  
      }  
    ]  
  }  
]  
}
```

## 4.9 监控

### 4.9.1 获取实时监控数据

此操作用来获取 HBlock 的实时性能数据。

#### ● 请求语法

```
GET
/rest/v1/system/monitor/current?dimension=dimension&instanceId=instanceId&metric=metric
HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

参数	描述	是否必须
dimension	监控对象。 类型：枚举。 取值： <ul style="list-style-type: none"> <li>● system: 系统。</li> <li>● server: 服务器。</li> <li>● disk: 数据目录所在磁盘分区。</li> <li>● LUN: 卷。</li> </ul> 默认值为 system。	否
instanceId	监控对象实例的唯一标识，可以填写多个，以英文逗号隔开。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● 监控对象为 system，没有实例。</li> <li>● 监控对象为 server，实例取值为服务器 ID。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● 如果监控对象为 <code>disk</code>，实例取值为“<code>serverId+路径名称</code>”，格式为 <code>serverId:/diskpath</code>。</li> <li>● 监控对象为 <code>LUN</code>，实例取值为卷名称。</li> </ul> <p>如果不填写，默认查看监控对象下的所有实例性能数据。</p>	
<b>metric</b>	<p>指标名称，可以填写多个，以英文逗号隔开。指标具体描述详见<a href="#">监控指标</a>。</p> <p>类型：字符串</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● 监控对象为 <code>system</code>:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> </ul> </li> <li>● 监控对象为 <code>server</code>:                             <ul style="list-style-type: none"> <li>■ CPU_Rate</li> <li>■ Mem_Rate</li> </ul> </li> </ul>	否

	<ul style="list-style-type: none"> <li>■ Mem_Total</li> <li>■ Mem_Used</li> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> <li>● 监控对象为 disk:                             <ul style="list-style-type: none"> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> </ul> </li> <li>● 监控对象为 LUN:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> </ul> </li> </ul>	
--	--	--

	<ul style="list-style-type: none"> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> </ul> <p>不填写，默认查询监控对象的全部指标。</p>	
--	--	--

● 响应结果

名称	描述
monitoring	监控信息集合，包含：dimension、instanceId、metric、avgValueSeries。 类型：数组
dimension	监控对象： <ul style="list-style-type: none"> <li>● system：系统。</li> <li>● server：服务器。</li> <li>● disk：数据目录所在磁盘分区。</li> <li>● LUN：卷。</li> </ul>
instanceId	监控对象实例： <ul style="list-style-type: none"> <li>● 监控对象为 system，显示 HBlock 名称。</li> <li>● 监控对象为 server，显示为服务器 ID。</li> <li>● 如果监控对象为 disk，显示为数据目录。</li> <li>● 监控对象为 LUN，显示为卷名称。</li> </ul>
metric	指标名称。具体详见 <a href="#">监控指标</a> 。
avgValueSeries	监控数据的[时间,数值]序列。 时间为 unix 时间戳，精确到毫秒。

- 请求示例

查询服务器 hblock\_1、hblock\_2 的 CPU 使用率和内存使用率。

```
GET
/rest/v1/system/monitor/current?dimension=server&instanceId=hblock_1,hblock_2&metric=CPU_Rate,Mem_Rate HTTP/1.1
Date: Fri, 12 Aug 2022 07:48:25 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 2e40eab6db8440b5b5ec68f2dd5f8cf1
Connection: keep-alive
Content-Length: 455
Date: Fri, 12 Aug 2022 07:48:25 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "monitoring": [
      {
        "dimension": "server",
        "instanceId": "hblock_1",
        "metric": "CPU_Rate",
        "avgValueSeries": [
          [
            1660290480000,
            1.35
          ]
        ]
      },
      {
```

```
    "dimension": "server",
    "instanceId": "hblock_1",
    "metric": "Mem_Rate",
    "avgValueSeries": [
      [
        1660290480000,
        98.15
      ]
    ]
  },
  {
    "dimension": "server",
    "instanceId": "hblock_2",
    "metric": "CPU_Rate",
    "avgValueSeries": [
      [
        1660290480000,
        0.94
      ]
    ]
  },
  {
    "dimension": "server",
    "instanceId": "hblock_2",
    "metric": "Mem_Rate",
    "avgValueSeries": [
      [
        1660290480000,
        97.78
      ]
    ]
  }
]
}
```

## 4.9.2 获取历史性能数据

此操作用来获取 HBlock 的历史性能数据。

### ● 请求语法

```
GET
/rest/v1/system/monitor?startTime=startTime&endTime=endTime&dimension=dimension&instanceId=instanceId1&metric=metric1,metric2 HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
startTime	查询起始时间。 类型：时间戳 取值：unix 时间戳（UTC），精确到毫秒。默认值为结束时间 2 小时之前的时间点。 <b>注意：</b> 起始时间必须早于结束时间，且起始时间不能早于服务器当前时间一年。	否
endTime	查询结束时间。 类型：时间戳 取值：unix 时间戳（UTC），精确到毫秒。默认值为当前时间。 <b>注意：</b> 起始时间必须早于结束时间。	否
dimension	监控对象。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● system: 系统。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● <b>server</b>: 服务器。</li> <li>● <b>disk</b>: 数据目录所在磁盘分区。</li> <li>● <b>LUN</b>: 卷。</li> </ul> 默认值为 <b>system</b> 。	
<b>instanceId</b>	监控对象实例的唯一标识。 类型: 字符串 取值: <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>, 没有实例。</li> <li>● 监控对象为 <b>server</b>, 必填, 实例取值为服务器 ID。</li> <li>● 如果监控对象为 <b>disk</b>, 必填, 实例取值为 “<i>serverId+路径名称</i>”, 格式为 <i>serverId:/diskpath</i>。</li> <li>● 监控对象为 <b>LUN</b>, 必填, 实例取值为卷名称。</li> </ul>	否
<b>metric</b>	指标名称, 可以填写多个, 以英文逗号隔开。指标具体描述详见 <a href="#">监控指标</a> 。 类型: 字符串 取值: <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Path_Cap</li> </ul> </li> </ul>	否

	<ul style="list-style-type: none"> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> <li>● 监控对象为 server:                             <ul style="list-style-type: none"> <li>■ CPU_Rate</li> <li>■ Mem_Rate</li> <li>■ Mem_Total</li> <li>■ Mem_Used</li> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> <li>■ Path_Cap</li> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> </ul> </li> <li>● 监控对象为 disk:                             <ul style="list-style-type: none"> <li>■ Path_Cap</li> </ul> </li> </ul>	
--	---	--

	<ul style="list-style-type: none"> <li>■ Path_Used</li> <li>■ Path_Rate</li> <li>■ Path_Cap_Quota</li> <li>■ Path_Cap_Quota_Used</li> <li>■ Path_Cap_Quota_Rate</li> <li>● 监控对象为 LUN:                             <ul style="list-style-type: none"> <li>■ IOPS</li> <li>■ R_IOPS</li> <li>■ W_IOPS</li> <li>■ Bandwidth</li> <li>■ R_Bandwidth</li> <li>■ W_Bandwidth</li> <li>■ Latency</li> <li>■ W_Latency</li> <li>■ R_Latency</li> </ul> </li> </ul> <p>不填写，默认获取监控对象的全部指标。</p>	
--	--	--

● 响应结果

名称	描述
monitoring	监控信息集合，包含：dimension、instanceId、metric、avgValueSeries。 类型：数组
dimension	监控对象： <ul style="list-style-type: none"> <li>● system: 系统。</li> <li>● server: 服务器。</li> <li>● disk: 数据目录所在磁盘分区。</li> <li>● LUN: 卷。</li> </ul>

<b>instanceId</b>	监控对象实例： <ul style="list-style-type: none"> <li>● 监控对象为 <b>system</b>，显示 HBlock 名称。</li> <li>● 监控对象为 <b>server</b>，显示为服务器 ID。</li> <li>● 如果监控对象为 <b>disk</b>，显示为“<b>serverId+路径名称</b>”，格式为 <b>serverId:/path/dir</b>。</li> <li>● 监控对象为 <b>LUN</b>，显示为卷名称。</li> </ul>
<b>metric</b>	指标名称。具体详见 <a href="#">监控指标</a> 。
<b>avgValueSeries</b>	监控数据的[时间,数值]序列。 时间为 <b>unix</b> 时间戳，精确到毫秒。

### ● 请求示例

查询卷 lun01a 2022-08-12 13:43:57 至 2022-08-12 14:03:57 的 IOPS 和带宽。

```

GET
/rest/v1/system/monitor?startTime=1660283037000&endTime=1660284237000&dimension=LUN&instanceId=lun01a&metric=IOPS,Bandwidth HTTP/1.1
Date: Fri, 12 Aug 2022 06:07:39 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
    
```

### ● 响应示例

```

HTTP/1.1 200 OK
x-hblock-request-id: bdbfd372ebf143ae8c08c79b09909fc9
Connection: keep-alive
Content-Length: 2464
Date: Fri, 12 Aug 2022 06:07:39 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "monitoring": [
    
```

```
{
  "dimension": "LUN",
  "instanceId": "lun01a",
  "metric": "IOPS",
  "avgValueSeries": [
    [
      1660283020000,
      0
    ],
    [
      1660283040000,
      0
    ],
    [
      1660283060000,
      0
    ],
    .....
    [
      1660284120000,
      0
    ],
    [
      1660284140000,
      0
    ],
    [
      1660284160000,
      0.45
    ],
    [
      1660284180000,
      304.75
    ]
  ]
}
```

```
    ],  
    [  
      1660284200000,  
      108  
    ],  
    [  
      1660284220000,  
      141.9  
    ],  
    [  
      1660284240000,  
      84.8  
    ]  
  ]  
},  
{  
  "dimension": "LUN",  
  "instanceId": "lun01a",  
  "metric": "Bandwidth",  
  "avgValueSeries": [  
    [  
      1660283020000,  
      0  
    ],  
    [  
      1660283040000,  
      0  
    ],  
    [  
      1660283060000,  
      0  
    ],  
    .....  
  ]  
}
```

```
[
  1660284080000,
  0
],
[
  1660284100000,
  0
],
[
  1660284120000,
  0
],
[
  1660284140000,
  0
],
[
  1660284160000,
  1843
],
[
  1660284180000,
  79475506
],
[
  1660284200000,
  28099583
],
[
  1660284220000,
  37126348
],
[
  1660284240000,
  22083788
]
```



### 4.9.3 错误码列表

HTTP status	错误码	错误信息	描述
400	ArgumentNotAllowed	The argument ' <i>argument</i> ' is not allowed when value at ' <i>argument2</i> ' is <i>value2</i> .	参数值不正确。
400	ExceedQueryRange	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must be <i>message</i> .	超出限定范围，无法执行该操作。
400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [ <i>value1</i> , <i>value2</i> ... ]	枚举参数不合法。
400	InvalidTime	The end time must be later than the start time.	开始时间必须早于结束时间
400	InvalidTimeFormat	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	时间格式不合法。

## 4.10 告警

**注意：**告警状态为告警中的数据存储上限为 10000 条，达到上限后，新的告警无法触发。

届时请尽快解决相关故障，或者尝试手动解除告警。

### 4.10.1 批量查询 HBlock 告警信息

此操作用来查询 HBlock 告警信息。

#### ● 请求语法

```
GET /rest/v1/system/alarm?alarmStatus=alarmStatus&filter=filter&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

名称	描述	是否必须
alarmStatus	告警状态： <ul style="list-style-type: none"> <li>● Unresolved: 告警中。</li> <li>● Resolved: 告警已解除。</li> <li>● Expired: 告警已失效。</li> </ul> 默认值为 Unresolved。 说明：已解除、已失效数据仅支持查询最近 10000 条。告警中的数据超出 10000 条告警，功能将不可用。	否
filter	设置查询的过滤条件。可以选择多个查询，如果是或的关系，使用 or 将查询条件分隔开；如果是与的关系，使用 and 将查询条件分隔开。 支持的查询类型包括： <ul style="list-style-type: none"> <li>● instanceSnapshot: 告警实例的详情。</li> <li>● alarmRule: 告警规则编码：</li> </ul>	否

	<ul style="list-style-type: none"> <li>➤ PathIOError。</li> <li>➤ DiskWriteSlow。</li> <li>➤ LicenseWillExpire。</li> <li>➤ LicenseExpired。</li> <li>➤ LicenseMaintenanceWillExpire。</li> <li>➤ LicenseMaintenanceExpired。</li> <li>➤ TrialVersionWillExpire。</li> <li>➤ ResourceUsageApproachingLimit。</li> <li>➤ AlarmNumberApproachingLimit。</li> <li>➤ FailToSendAlarmEmail。</li> <li>➤ DiskPathHealthStatusWarning。</li> <li>➤ DiskPathHealthStatusError。</li> <li>➤ DataServiceHealthStatusWarning。</li> <li>➤ DataServiceHealthStatusError。</li> <li>➤ ProtocolServiceAbnormal。</li> <li>➤ FaultDomainWarning。</li> <li>➤ FaultDomainError。</li> <li>➤ CapacityQuotaUsageExceedsThreshold。</li> <li>➤ CapacityQuotaUsageApproachLimit。</li> <li>➤ DiskUsageExceedsThreshold。</li> <li>➤ InsufficientFDForLUNToWrite。</li> <li>● severity: 告警级别:                         <ul style="list-style-type: none"> <li>➤ Warning: 警告。</li> <li>➤ Major: 重要。</li> <li>➤ Critical: 严重。</li> </ul> </li> <li>● muteStatus: 静默状态:                         <ul style="list-style-type: none"> <li>➤ Normal: 正常。</li> </ul> </li> </ul>	
--	---	--

	<ul style="list-style-type: none"> <li>➤ Muted: 静默。</li> <li>● alarmTime: 告警发生时间, 通过&gt;和&lt;支持范围查询。unix 时间戳 (UTC), 精确到毫秒。</li> <li>● resolveTime: 解除时间, 通过&gt;和&lt;支持范围查询。unix 时间戳 (UTC), 精确到毫秒。</li> <li>● resolveType: 告警解除方式:                         <ul style="list-style-type: none"> <li>➤ Manual: 手动解除警告。</li> <li>➤ Auto: 自动解除告警。</li> </ul> </li> <li>● expireTime: 告警失效时间, 通过&gt;和&lt;支持范围查询。unix 时间戳 (UTC), 精确到毫秒。</li> </ul>	
range	查询范围, 格式: i-j, 其中 i,j 是正整数, i 小于 j, j-i<1000。 按照 alarmTime 进行逆序排序, 返回第 i 条到第 j 条的告警。 如果输入的查询范围超过告警的总个数, 那么返回空结果集。 如果不输入 range 参数, 那么返回所有符合条件的告警。	否

### ● 响应结果

名称	描述
totalCount	符合查询条件的告警条数。
alarmList	告警列表, 包含: alarmId、instanceId、instanceSnapshot、alarmRule、severity、alarmStatus、duration、alarmTime、muteStatus、alarmValue、currentValue、resolveTime、resolveValue、resolveType、expireTime、reason、muteDueTime、muteOperations。 类型: 数组
alarmId	告警 ID。
instanceId	告警实例 ID。
instanceSnapshot	告警实例快照, 即告警发生时告警实例的详细信息。

alarmRule	告警规则。
severity	告警级别： <ul style="list-style-type: none"> <li>● <b>Warning</b>: 警告。</li> <li>● <b>Major</b>: 重要。</li> <li>● <b>Critical</b>: 严重。</li> </ul>
alarmStatus	告警状态： <ul style="list-style-type: none"> <li>● <b>Unresolved</b>: 告警中。</li> <li>● <b>Resolved</b>: 告警已解除。</li> <li>● <b>Expired</b>: 告警已失效。</li> </ul>
duration	告警持续时长。unix 时间戳（UTC），精确到毫秒。
alarmTime	告警发生时间。unix 时间戳（UTC），精确到毫秒。
muteStatus	静默状态： <ul style="list-style-type: none"> <li>● <b>Muted</b>: 静默。</li> <li>● <b>Normal</b>: 正常。</li> </ul>
alarmValue	告警时数值。仅有数值型指标对应的告警会有返回此项，以下告警规则会涉及数值型指标： <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit</b>: 告警中的告警条数接近上限，显示告警中数据容量使用率，百分比，单位是%。计算公式：<math>(\text{告警中的告警总条数} / \text{告警中条数上限}) * 100\%</math>。</li> <li>● <b>ResourceUsageApproachingLimit</b>: 资源用量接近使用上限，显示许可证可用容量使用率，百分比，单位是%。计算公式：<math>(\text{本地卷总容量} / \text{许可证允许的容量}) * 100\%</math>。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold</b>: 配额使用率超阈值，显示系统、服务器或数据目录关联磁盘的配额使用率（<code>Path_Cap_Quota_Rate</code>），百分数，单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit</b>: 配额用尽，显示系统、服务器或数据目录关联磁盘的配额使用率</li> </ul>

	<p>(Path_Cap_Quota_Rate)，百分数，单位是%。</p> <ul style="list-style-type: none"> <li>● <b>DiskUsageExceedsThreshold</b>: 磁盘使用率超阈值，系统、服务器或数据目录关联磁盘的使用率 (Path_Rate)，百分数，单位是%。</li> </ul>
currentValue	<p>当前数值。仅有数值型指标对应的告警会有返回此项, 以下告警规则会涉及数值型指标:</p> <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit</b>: 告警中的告警条数接近上限，显示告警中数据容量使用率，百分比，单位是%。计算公式：<math>(\text{告警中的告警总条数}/\text{告警中条数上限}) * 100\%</math>。</li> <li>● <b>ResourceUsageApproachingLimit</b>: 资源用量接近使用上限，显示许可证可用容量使用率，百分比，单位是%。计算公式：<math>(\text{本地卷总容量}/\text{许可证允许的容量}) * 100\%</math>。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold</b>: 配额使用率超阈值，显示系统、服务器或数据目录关联磁盘的配额使用率 (Path_Cap_Quota_Rate)，百分数，单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit</b>: 配额用尽，显示系统、服务器或数据目录关联磁盘的配额使用率 (Path_Cap_Quota_Rate)，百分数，单位是%。</li> <li>● <b>DiskUsageExceedsThreshold</b>: 磁盘使用率超阈值，系统、服务器或数据目录关联磁盘的使用率 (Path_Rate)，百分数，单位是%。</li> </ul>
resolveTime	<p>告警解除时间。unix 时间戳 (UTC)，精确到毫秒。</p> <p>Resolved 状态的告警会返回此字段。</p>
resolveValue	<p>解除告警时的数值。仅有数值型指标对应的告警会有返回此项:</p> <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit</b>: 告警中的告警条数接近上限，显示告警中数据容量使用率，百分比，单位是%。计算公式：<math>(\text{告警中的告警总条数}/\text{告警中条数上限}) * 100\%</math>。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>ResourceUsageApproachingLimit</b>: 资源用量接近使用上限，显示许可证可用容量使用率，百分比，单位是%。计算公式： (本地卷总容量/许可证允许的容量)*100%。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold</b>: 配额使用率超阈值，显示系统、服务器或数据目录关联磁盘的配额使用率 (Path_Cap_Quota_Rate)，百分数，单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit</b>: 配额用尽，显示系统、服务器或数据目录关联磁盘的配额使用率 (Path_Cap_Quota_Rate)，百分数，单位是%。</li> <li>● <b>DiskUsageExceedsThreshold</b>: 磁盘使用率超阈值，系统、服务器或数据目录关联磁盘的使用率 (Path_Rate)，百分数，单位是%。</li> </ul>
resolveType	告警解除方式： <ul style="list-style-type: none"> <li>● <b>Auto</b>: 自动解除。</li> <li>● <b>Manual</b>: 手动解除。</li> </ul>
expireTime	告警失效时间。 Expired 状态的告警会返回此字段。
reason	告警解除原因或告警失效原因。
muteDueTime	静默截止时间，unix 时间戳 (UTC)，精确到毫秒。 如果多次静默，返回最后一次操作的静默截止时间。
muteOperations	静默操作，包含: operTime、operType、reason、dueTime。 类型: 数组
operTime	静默操作时间。
operType	静默操作类型： <ul style="list-style-type: none"> <li>● <b>Mute</b>: 静默。</li> <li>● <b>AutoUnmute</b>: 自动解除静默。</li> <li>● <b>ManualUnmute</b>: 手动解除静默。</li> </ul>

reason	静默/解除静默的原因。
dueTime	静默截止时间。

- 请求示例 1

查询告警中的告警信息。

```
GET /rest/v1/system/alarm?alarmStatus=Unresolved HTTP/1.1
Date: Mon, 8 Jan 2024 03:26:58 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature
```

- 响应示例 1

```
HTTP/1.1 200 OK
Date: Mon, 8 Jan 2024 03:26:58 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 1448
Connection: keep-alive
x-hblock-request-id: 0ba4372189a8470ca43fb8bea7959bf5
Server: HBlock

{
  "data": {
    "totalCount": 5,
    "alarmList": [
      {
        "alarmId": "4jlSnwld",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3:/mnt/stor",
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 346407,
        "alarmTime": 1704684071663,
        "alarmValue": "93.69",
```

```
        "currentValue": "92.63",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4jlSnJ9h",
        "instanceId": "hblock_3",
        "instanceSnapshot": "hblock_3",
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 346861,
        "alarmTime": 1704684071209,
        "alarmValue": "93.69",
        "currentValue": "92.63",
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4jlSmXuZ",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
        "alarmRule": "FaultDomainWarning",
        "severity": "Warning",
        "alarmStatus": "Unresolved",
        "duration": 347413,
        "alarmTime": 1704684070657,
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4jlSmPPo",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
        "alarmRule": "DataServiceHealthStatusWarning",
        "severity": "Warning",
```

```

        "alarmStatus": "Unresolved",
        "duration": 347608,
        "alarmTime": 1704684070462,
        "muteStatus": "Normal"
    },
    {
        "alarmId": "4j1htUe1",
        "instanceId": "hblock_4",
        "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
        "alarmRule": "ProtocolServiceAbnormal",
        "severity": "Major",
        "alarmStatus": "Unresolved",
        "duration": 2634530,
        "alarmTime": 1704681783540,
        "muteStatus": "Normal"
    }
]
}

```

### ● 请求示例 2

查询已解除的告警信息。

```

GET /rest/v1/system/alarm?alarmStatus=Resolved HTTP/1.1
Date: Mon, 8 Jan 2024 05:34:23 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature

```

### ● 响应示例 2

```

HTTP/1.1 200 OK
Date: Mon, 8 Jan 2024 05:34:23 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 3268
Connection: keep-alive
x-hblock-request-id: b47f7f2aae794363bc577eb304425122

```

Server: HBlock

```
{
  "data": {
    "totalCount": 10,
    "alarmList": [
      {
        "alarmId": "4jlSnwld",
        "instanceId": "hblock_3:/mnt/stor",
        "instanceSnapshot": "hblock_3:/mnt/stor",
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",
        "severity": "Warning",
        "alarmStatus": "Resolved",
        "duration": 603073,
        "alarmTime": 1704684071663,
        "alarmValue": "93.69",
        "resolveTime": 1704684674736,
        "resolveValue": "5.06",
        "resolveType": "Auto",
        "muteStatus": "Normal"
      },
      {
        "alarmId": "4jlSnJ9h",
        "instanceId": "hblock_3",
        "instanceSnapshot": "hblock_3",
        "alarmRule": "CapacityQuotaUsageExceedsThreshold",
        "severity": "Warning",
        "alarmStatus": "Resolved",
        "duration": 603023,
        "alarmTime": 1704684071209,
        "alarmValue": "93.69",
        "resolveTime": 1704684674232,
        "resolveValue": "5.06",
        "resolveType": "Auto",
        "muteStatus": "Normal"
      }
    ]
  }
}
```

```
  },
  {
    "alarmId": "4jlSmXuZ",
    "instanceId": "hblock_3:/mnt/stor",
    "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
    "alarmRule": "FaultDomainWarning",
    "severity": "Warning",
    "alarmStatus": "Resolved",
    "duration": 603430,
    "alarmTime": 1704684070657,
    "resolveTime": 1704684674087,
    "resolveType": "Auto",
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4jlSmPPo",
    "instanceId": "hblock_3:/mnt/stor",
    "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102:/mnt/stor",
    "alarmRule": "DataServiceHealthStatusWarning",
    "severity": "Warning",
    "alarmStatus": "Resolved",
    "duration": 602999,
    "alarmTime": 1704684070462,
    "resolveTime": 1704684673461,
    "resolveType": "Auto",
    "muteStatus": "Normal"
  },
  {
    "alarmId": "4ilZH7wX",
    "instanceId": "hblock_1",
    "instanceSnapshot": "hblock_1,hblockserver,192.168.0.110",
    "alarmRule": "ProtocolServiceAbnormal",
    "severity": "Major",
    "alarmStatus": "Resolved",
    "duration": 230649424,
```

```
"alarmTime": 1704446197387,
"resolveTime": 1704676846811,
"resolveType": "Auto",
"muteStatus": "Normal"
},
{
  "alarmId": "4hDwJnaH",
  "instanceId": "hblock_3",
  "instanceSnapshot": "hblock_3,ecs-9689-0915140,192.168.0.102",
  "alarmRule": "ProtocolServiceAbnormal",
  "severity": "Major",
  "alarmStatus": "Resolved",
  "duration": 361874,
  "alarmTime": 1704275265260,
  "resolveTime": 1704275627134,
  "resolveType": "Auto",
  "muteStatus": "Normal"
},
{
  "alarmId": "4hD1BMWS",
  "instanceId": "hblock_4",
  "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
  "alarmRule": "ProtocolServiceAbnormal",
  "severity": "Major",
  "alarmStatus": "Resolved",
  "duration": 2291572,
  "alarmTime": 1704273335042,
  "resolveTime": 1704275626614,
  "resolveType": "Auto",
  "muteStatus": "Normal"
},
{
  "alarmId": "4hD2AgSt",
  "instanceId": "hblock_1:/mnt/stor",
  "instanceSnapshot": "hblock_1,hblockserver,192.168.0.110:/mnt/stor",
```

```
"alarmRule": "FaultDomainError",
"severity": "Major",
"alarmStatus": "Resolved",
"duration": 61433,
"alarmTime": 1704273396083,
"resolveTime": 1704273457516,
"resolveType": "Auto",
"muteStatus": "Normal"
},
{
  "alarmId": "4hD2zESY",
  "instanceId": "hblock_1:/mnt/stor",
  "instanceSnapshot": "hblock_1,hblockserver,192.168.0.110:/mnt/stor",
  "alarmRule": "DataServiceHealthStatusError",
  "severity": "Major",
  "alarmStatus": "Resolved",
  "duration": 61164,
  "alarmTime": 1704273395799,
  "resolveTime": 1704273456963,
  "resolveType": "Auto",
  "muteStatus": "Normal"
},
{
  "alarmId": "4hCtBCK5",
  "instanceId": "hblock_4:/mnt/storage01",
  "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202:/mnt/storage01",
  "alarmRule": "DataServiceHealthStatusError",
  "severity": "Major",
  "alarmStatus": "Resolved",
  "duration": 60395,
  "alarmTime": 1704271227603,
  "resolveTime": 1704271287998,
  "resolveType": "Auto",
  "muteStatus": "Normal"
}
```

```
    ]  
  }  
}
```

### ● 请求示例 3

查询告警中、告警级别为 Major 的告警信息。

```
GET /rest/v1/system/alarm?alarmStatus=Unresolved&filter=severity:Major HTTP/1.1  
Date: Mon, 8 Jan 2024 05:38:29 GMT  
Host: 192.168.0.1110:1443  
Authorization: HBlock userName:signature
```

### ● 响应示例 3

```
HTTP/1.1 200 OK  
Date: Mon, 8 Jan 2024 05:38:29 GMT  
Content-Type: application/json; charset=utf-8  
Content-Length: 299  
Connection: keep-alive  
x-hblock-request-id: 988b145267ac41c886954b6f81598ec4  
Server: HBlock  
  
{  
  "data": {  
    "totalCount": 1,  
    "alarmList": [  
      {  
        "alarmId": "4j1htUe1",  
        "instanceId": "hblock_4",  
        "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",  
        "alarmRule": "ProtocolServiceAbnormal",  
        "severity": "Major",  
        "alarmStatus": "Unresolved",  
        "duration": 10526372,  
        "alarmTime": 1704681783540,  
        "muteStatus": "Normal"  
      }  
    ]  
  }  
}
```

```
}  
  ]  
}  
}
```

## 4.10.2 查询指定的 HBlock 告警

此操作用来查询指定的 HBlock 告警信息。

### ● 请求语法

```
GET /rest/v1/system/alarm/alarmId HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

名称	描述	是否必须
alarmId	告警 ID。 仅 Unresolved 状态的告警支持单条查询。	是

### ● 响应结果

名称	描述
alarmId	告警 ID。
instanceId	告警实例 ID。
instanceSnapshot	告警实例快照，即告警发生时告警实例的详细信息。
alarmRule	告警规则。
severity	告警级别： <ul style="list-style-type: none"> <li>● Warning: 警告。</li> <li>● Major: 重要。</li> <li>● Critical: 严重。</li> </ul>
alarmStatus	告警状态： <ul style="list-style-type: none"> <li>● Unresolved: 告警中。</li> <li>● Resolved: 告警已解除。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Expired:</b> 告警已失效。</li> </ul>
<b>duration</b>	告警持续时长。unix 时间戳 (UTC)，精确到毫秒。
<b>alarmTime</b>	告警发生时间。unix 时间戳 (UTC)，精确到毫秒。
<b>alarmValue</b>	<p>告警时数值。仅有数值型指标对应的告警会有数值显示，以下告警规则会涉及数值型指标：</p> <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit:</b> 告警中的告警条数接近上限，显示告警中数据容量使用率，百分比，单位是%。计算公式：<math>(\text{告警中的告警总条数} / \text{告警中条数上限}) * 100\%</math>。</li> <li>● <b>ResourceUsageApproachingLimit:</b> 资源用量接近使用上限，显示许可证可用容量使用率，百分比，单位是%。计算公式：<math>(\text{本地卷总容量} / \text{许可证允许的容量}) * 100\%</math>。</li> <li>● <b>CapacityQuotaUsageExceedsThreshold:</b> 配额使用率超阈值，显示系统、服务器或数据目录关联磁盘的配额使用率 (<math>\text{Path\_Cap\_Quota\_Rate}</math>)，百分数，单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit:</b> 配额用尽，显示系统、服务器或数据目录关联磁盘的配额使用率 (<math>\text{Path\_Cap\_Quota\_Rate}</math>)，百分数，单位是%。</li> <li>● <b>DiskUsageExceedsThreshold:</b> 磁盘使用率超阈值，系统、服务器或数据目录关联磁盘的使用率 (<math>\text{Path\_Rate}</math>)，百分数，单位是%。</li> </ul>
<b>currentValue</b>	<p>当前数值。仅有数值型指标对应的告警会有数值显示，以下告警规则会涉及数值型指标：</p> <ul style="list-style-type: none"> <li>● <b>AlarmNumberApproachingLimit:</b> 告警中的告警条数接近上限，显示告警中数据容量使用率，百分比，单位是%。计算公式：<math>(\text{告警中的告警总条数} / \text{告警中条数上限}) * 100\%</math>。</li> <li>● <b>ResourceUsageApproachingLimit:</b> 资源用量接近使用上限，显示许可证可用容量使用率，百分比，单位是%。计算公式：</li> </ul>

	<p>(本地卷总容量/许可证允许的容量)*100%。</p> <ul style="list-style-type: none"> <li>● <b>CapacityQuotaUsageExceedsThreshold</b>: 配额使用率超阈值, 显示系统、服务器或数据目录关联磁盘的配额使用率 (Path_Cap_Quota_Rate), 百分数, 单位是%。</li> <li>● <b>CapacityQuotaUsageApproachLimit</b>: 配额用尽, 显示系统、服务器或数据目录关联磁盘的配额使用率 (Path_Cap_Quota_Rate), 百分数, 单位是%。</li> <li>● <b>DiskUsageExceedsThreshold</b>: 磁盘使用率超阈值, 系统、服务器或数据目录关联磁盘的使用率 (Path_Rate), 百分数, 单位是%。</li> </ul>
muteStatus	<p>静默状态:</p> <ul style="list-style-type: none"> <li>● <b>Muted</b>: 静默。</li> <li>● <b>Normal</b>: 正常。</li> </ul>
muteDueTime	<p>静默截止时间, unix 时间戳 (UTC), 精确到毫秒。 如果多次静默, 返回最后一次操作的静默截止时间。</p>
muteOperations	<p>静默操作, 包含: operTime、operType、reason、dueTime。 类型: 数组</p>
operTime	<p>静默操作时间。</p>
operType	<p>静默操作类型:</p> <ul style="list-style-type: none"> <li>● <b>Mute</b>: 静默。</li> <li>● <b>AutoUnmute</b>: 自动解除静默。</li> <li>● <b>ManualUnmute</b>: 手动解除静默。</li> </ul>
reason	<p>静默/解除静默的原因。</p>
dueTime	<p>静默截止时间。</p>

- 请求示例

查询告警 ID 为 4jlhtUe1 的告警信息。

```
GET /rest/v1/system/alarm/4j1htUe1 HTTP/1.1
Date: Mon, 8 Jan 2024 05:42:22 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature
```

### ● 响应示例

```
HTTP/1.1 200 OK
Date: Mon, 8 Jan 2024 05:42:22 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 268
Connection: keep-alive
x-hblock-request-id: dea2f8c199624d33aed3c4daa4654c2a
Server: HBlock

{
  "data": {
    "alarmId": "4j1htUe1",
    "instanceId": "hblock_4",
    "instanceSnapshot": "hblock_4,ecs-9689-0915141,192.168.0.202",
    "alarmRule": "ProtocolServiceAbnormal",
    "severity": "Major",
    "alarmStatus": "Unresolved",
    "duration": 10718615,
    "alarmTime": 1704681783540,
    "muteStatus": "Normal"
  }
}
```

### 4.10.3 手动解除 HBlock 告警

此操作用来手动解除指定的 HBlock 告警。

- 请求语法

```
PUT /rest/v1/system/alarm/alarmId/resolve HTTP/1.1
Date: date
Host: ip:port
Content-Type: text/plain
Content-Length: length
Authorization: authorization

{
    "reason": reason
}
```

- 请求参数

名称	描述	是否必须
alarmId	告警 ID。	是
reason	解除告警的原因。 类型：字符串 取值：1~50 位字符串。	是

- 请求示例

手动解除告警 1kyvM3zJ。

```
PUT /rest/v1/system/alarm/1kyvM3zJ/resolve HTTP/1.1
Date: Wed, 17 Aug 2022 06:29:20 GMT
Host: 192.168.0.121:1443
Content-Type: text/plain
Content-Length: 41
```

```
Authorization: HBlock userName:signature
```

```
{  
  "reason": "已申请新 license"  
}
```

- 响应示例

```
HTTP/1.1 204 No Content  
Date: Wed, 17 Aug 2022 06:29:21 GMT  
Connection: keep-alive  
x-hblock-request-id: 9fe8d7733bd34c40a8895d674162c072  
Server: HBlock
```

#### 4.10.4 静默告警/解除告警静默

此操作用来静默警或者解除告警静默。

##### ● 请求语法

```
PUT /rest/v1/system/alarm/alarmId/mute HTTP/1.1
```

Date: *date*

Host: *ip:port*

Content-Type: text/plain

Content-Length: *length*

Authorization: *authorization*

```
{
  "operType": operType,
  "dueTime": dueTime,
  "reason": reason
}
```

##### ● 请求参数

名称	描述	是否必须
alarmId	告警 ID。	是
operType	静默操作。 类型：字符串 取值： ● Mute: 静默告警。 ● ManualUnmute: 解除告警静默。	是
dueTime	静默截止时间。 类型：时间戳 取值：unix 时间戳（UTC），精确到毫秒。如果静默操作为 Mute，此项必填。	否

reason	静默/解除静默的原因。 类型：字符串 取值：1~50 位字符串。	是
--------	--	---

● 请求示例 1

静默告警 1kUyXPFy。

```

PUT /rest/v1/system/alarm/1kUyXPFy/mute HTTP/1.1
Date: Thu, 18 Aug 2022 08:25:06 GMT
Host: 192.168.0.121:1443
Content-Type: text/plain
Content-Length: 90
Authorization: HBlock userName:signature

{
  "operType": "Mute",
  "dueTime": 1660903200000,
  "reason": "renew license"
}
    
```

● 响应示例 1

```

HTTP/1.1 204 No Content
Date: Thu, 18 Aug 2022 08:25:06 GMT
Connection: keep-alive
x-hblock-request-id: 40d4c09de42d4d4cbf2f63e42d929208
Server: HBlock
    
```

● 请求示例 2

解除告警 1kUyXPFy 的静默。

```

PUT /rest/v1/system/alarm/1kUyXPFy/mute HTTP/1.1
Date: Thu, 18 Aug 2022 08:35:55 GMT
Host: 192.168.0.121:1443
Content-Type: text/plain
    
```

```
Content-Length: 81
Authorization: HBlock userName:signature

{
  "operType":"ManualUnmute",
  "reason": "license is about to expire."
}
```

- 响应示例

```
HTTP/1.1 204 No Content
Date: Thu, 18 Aug 2022 08:35:55 GMT
Connection: keep-alive
x-hblock-request-id: 40d4c09de42d4d4cbf2f63e42d929208
Server: HBlock
```

## 4.10.5 导出告警

此操作用来导出告警信息。

### ● 请求语法

```
GET /rest/v1/system/alarm/file?alarmStatus=alarmStatus HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

名称	描述	是否必须
alarmStatus	告警状态。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● Unresolved：告警中。</li> <li>● Resolved：告警已解除。</li> <li>● Expired：告警已失效。</li> </ul> 默认值为 Unresolved。	否

### ● 请求示例

导出告警中的告警。

```
GET /rest/v1/system/alarm/file HTTP/1.1
Date: Mon, 08 Jan 2024 05:47:24 GMT
Host: 192.168.0.110:1443
Authorization: HBlock userName:signature
```

### ● 响应示例

```
HTTP/1.1 204 No Content
Date: Mon, 08 Jan 2024 05:47:24 GMT
```

```
Connection: keep-alive
x-hblock-request-id: 1c631d8b1b3549d7b4959d2bfae5f13b
Content-Disposition: attachment;filename=alarm_Unresolved_20220817160233.csv
Transfer-Encoding: chunked
Server: HBlock

alarmId,instanceId,instanceSnapshot,alarmRule,severity,alarmStatus,alarmTime,alarmValue
,muteStatus,muteOperations
4jlhtUe1,hblock_4,"hblock_4,ecs-9689-
0915141,192.168.0.202",ProtocolServiceAbnormal,Major,Unresolved,2024-01-08
10:43:03,,Normal,
```

### 4.10.6 错误码列表

HTTP status	错误码	错误信息	描述
400	InvalidArgumentLength	Value at 'reason' failed to satisfy constraint: Argument must not exceed 50 characters.	参数 reason 的长度不可超过 50 字符。
400	InvalidEnumValue	Value <i>value</i> at 'argument' failed to satisfy constraint: Argument must satisfy enum value set: [ <i>value1</i> , <i>value2</i> ... ]	枚举参数不合法。
400	InvalidFilterEnum	Value <i>value</i> at 'filter' failed to satisfy constraint: Argument can only contain: [instanceSnapshot, alarmRule, severity, muteStatus, alarmTime, resolveTime, resolveType, expireTime].	Filter 参数不正确。
400	MissingArgument	Value null at 'reason' failed to satisfy constraint: Argument must not be null.	静默与手动解除静默时，reason 字段必填。
400	MissingArgument	Value null at 'dueTime' failed to satisfy constraint: Argument must not be null.	静默操作时，dueTime 字段必填。
400	InvalidTimeFormat	Value <i>value</i> at 'argument' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	时间格式不合法。
400	NoSuchUnresolvedAlarm	The alarm with ID <i>ID</i> does not exist in 'Unresolved' alarm records.	指定告警 ID 的告警在告警记录中不存在。

## 4.11 事件和日志管理

事件名称描述详见附录用户事件列表和系统事件列表。

### 4.11.1 查看事件

此操作用来查看 HBlock 事件信息。

#### ● 请求语法

```
GET /rest/v1/system/event?number=number&type=type HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

参数	描述	是否必须
number	指定最近事件的查询个数。 类型：整型 取值：[1, 1000]，默认值为 1000。	否
type	指定事件的类型。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● user：用户事件。</li> <li>● system：系统事件。</li> </ul> 默认值为 user。	否

#### ● 响应结果

名称	描述
events	事件信息集合，包含：eventId、module、eventTime、name、

	requestId、requesterIP、statusCode、errorCode、errorMessage、instanceId。 类型：数组
eventId	事件 ID。
module	事件所属模块。
eventTime	对于用户事件：HBlock 接收到事件请求的时间，unix 时间戳（UTC），精确到毫秒。 对于系统事件：事件发生时间，unix 时间戳（UTC），精确到毫秒。
name	事件名称。 <b>说明：</b> 事件名称描述详见用户事件列表和系统事件列表。
requestId	用户事件请求 ID。
requesterIP	用户事件中，发起请求的源 IP 地址。
statusCode	用户事件响应状态码。
errorCode	用户事件错误码。 -：表示无错误码。
errorMessage	用户事件错误信息。 -：表示无错误信息。
detail.method	用户事件的请求方法。
detail.url	用户事件的 URL。
detail.body	事件的请求体。
instanceId	系统事件针对的实例 ID。 如果没有实例 ID，则不返回此项。

### ● 请求示例 1

查询 HBlock 用户事件。

```
GET /rest/v1/system/event?number=2 HTTP/1.1
Date: Thu, 04 Aug 2022 06:12:33 GMT
```

```
Authorization: HBlock userName:signature
```

```
Host:192.168.0.121:1443
```

### ● 响应示例 1

```
HTTP/1.1 200 OK
```

```
x-hblock-request-id: d701da5a23c94eef8a8efe0907b12751
```

```
Connection: keep-alive
```

```
Content-Length: 853
```

```
Date: Thu, 04 Aug 2022 06:12:33 GMT
```

```
Content-Type: application/json;charset=utf-8
```

```
Server: HBlock
```

```
{
  "data": {
    "events": [
      {
        "eventId": "78f85f5e-e8ca-42b9-889d-834a167283cc",
        "module": "System",
        "eventTime": 1659429529758,
        "name": "StartLogCollect",
        "requestId": "f630a474bc7e4d9eba75f3b9d28e5d51",
        "requesterIP": "36.111.88.33",
        "statusCode": 202,
        "errorCode": "-",
        "errorMessage": "-",
        "detail": {
          "method": "POST",
          "url": "/rest/v1/system/logcollect",
          "body":
            "{\"startTime\":1659422322714,\"endTime\":1659429522714,\"servers\":[\"hblock_1\", \"hblock_2\", \"hblock_3\"],\"logTypes\":[\"Config\", \"System\", \"Data\", \"Coordination\"]}"
        }
      },
      {
        "eventId": "37fd582e-6d36-4174-a9bd-94c825316f8a",
```

```
    "module": "System",
    "eventTime": 1659429493016,
    "name": "Login",
    "requestId": "ab3c00578efb4372b1fd622d827ad97c",
    "requesterIP": "36.111.88.33",
    "statusCode": 200,
    "errorCode": "-",
    "errorMessage": "-",
    "detail": {
      "method": "POST",
      "url": "/internal/v1/system/user/login",
      "body": "{\"userName\":\"storuser\"}"
    }
  }
]
```

### ● 请求示例 2

查询 HBlock 系统事件。

```
GET /rest/v1/system/event?number=2&type=system HTTP/1.1
Date: Thu, 4 Aug 2022 06:27:30 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 9199a40fdf4c48c58ec5a4f7a1acda95
Connection: keep-alive
Content-Length: 454
Date: Thu, 4 Aug 2022 06:27:30 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock
```

```
{
  "data": {
    "events": [
      {
        "eventId": "19cb57dd-5210-45af-a930-3d4cc70fe299",
        "module": "System",
        "eventTime": 1659343497897,
        "name": "DataResumed"
      },
      {
        "eventId": "1843075f-1281-4abc-af22-05a95c3a00ac",
        "module": "System",
        "eventTime": 1659343497897,
        "name": "DataLowRedundancy",
        "detail": {
          "body": "{\"percent\": \"0%\"}"
        }
      },
      {
        "eventId": "6b6d9ec4-4a20-44d2-b00c-f3f154373a11",
        "module": "Disk",
        "eventTime": 1659343495978,
        "name": "PathRemoved",
        "instanceId": "hblock_2:/mnt/storage02"
      }
    ]
  }
}
```

## 4.11.2 导出事件

此操作用来导出 HBlock 事件信息。

### ● 请求语法

```
GET /rest/v1/system/event/file?number=numebr&type=type HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 请求参数

参数	描述	是否必须
number	指定导出最近 HBlock 事件的个数。 说明：如果不指定，默认导出系统记录的所有 HBlock 事件，系统记录至少 6 个月的 HBlock 事件信息。 类型：整型 取值：[1, 10000]。	否
type	指定事件的类型。 类型：字符串 取值： <ul style="list-style-type: none"> <li>● user：用户事件。</li> <li>● system：系统事件。</li> </ul> 默认值为 user。	否

### ● 响应头

名称	描述
Content-Disposition	响应内容的显示形式。
Transfer-Encoding	实体传输给用户的编码形式。

- 请求示例 1

导出 HBlock 用户事件。

```
GET /rest/v1/system/event/file?number=3 HTTP/1.1
Date: Mon, 21 Mar 2022 01:53:48 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

- 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: 9fd643aa73bd4b60ab59a29e2904ab49
Connection: keep-alive
Date: Mon, 21 Mar 2022 01:53:48 GMT
Content-Disposition: attachment;filename=events_20220321095348.csv
Transfer-Encoding: chunked
Server: HBlock

eventId,eventTime,module,name,requestId,requesterIP,method,URL,body,statusCode,errorCode,errorMessage
a3830830-7f8a-40b2-a764-8ebfa6083f33,2022-03-21
09:53:28,LUN,ExpandLUN,5929bcbd0cf54e7f8a50b41f8bdd549b,111.198.231.68,PUT,http://218.7
8.41.185:1443/rest/v1/block/lun/lunName1/expand,"{
  ""capacity"":200000
}"
",400,LessThanCurrentCapacity,Value 200000 at 'capacity' failed to satisfy constraint:
Argument must be greater than 200000.
79a4c7b3-5b5c-46da-9725-f14237d462be,2022-03-21
09:53:21,LUN,SwitchLUN,a9be97449f434964960dbc8b48c05818,111.198.231.68,PUT,http://218.7
8.41.185:1443/rest/v1/block/lun/Lun1/switch,,200,-,-
9481763e-da71-4354-8a77-bff1c2d3effd,2022-03-21
09:53:09,LUN,SetLUN,9718a304ab1f4f03b9530ebeeef3b51fc,111.198.231.68,PUT,http://218.78.4
1.185:1443/rest/v1/block/lun/Lun1,"{
  ""config"":{
    ""writePolicy"":""WriteThrough""
```

```
}  
}  
",200,-,-
```

- 请求示例 2

导出 HBlock 系统事件。

```
GET /rest/v1/system/event/file?number=3&type=system HTTP/1.1  
Date: Thu, 4 Aug 2022 09:28:28 GMT  
Authorization: HBlock userName:signature  
Host:192.168.0.121:1443
```

- 响应示例 2

```
HTTP/1.1 200 OK  
x-hblock-request-id: 0de4cbbd38104d8aaeae7700e6af9bae  
Connection: keep-alive  
Date: Thu, 4 Aug 2022 09:28:28 GMT  
Content-Disposition: attachment;filename=events_20220321095348.csv  
Transfer-Encoding: chunked  
Server: HBlock  
  
eventId,eventTime,module,name,instanceId,details  
19cb57dd-5210-45af-a930-3d4cc70fe299,2022-08-01 16:44:57,System,DataResumed,null,{}  
1843075f-1281-4abc-af22-05a95c3a00ac,2022-08-01  
16:44:57,System,DataLowRedundancy,null,"{"percent": "0%"}"  
6b6d9ec4-4a20-44d2-b00c-f3f154373a11,2022-08-01  
16:44:55,Disk,PathRemoved,hblock_2:/mnt/storage02,{}
```

### 4.11.3 发起 HBlock 日志采集

此操作用来发起 HBlock 日志采集。

#### ● 请求语法

```
POST /rest/v1/system/logcollect HTTP/1.1
Date: date
Host: ip:port
Content-Length: length
Authorization: authorization

{
  "startTime": startTime,
  "endTime": endTime,
  "servers": [ server_ID ],
  "logTypes": [ LogTypes ],
  "outputDirectory": outputDirectory
}
```

#### ● 请求参数

参数	描述	是否必须
startTime	<p>HBlock 日志采集的起始时间。</p> <p>类型：时间戳</p> <p>取值：unix 时间戳（UTC），精确到毫秒。默认值为 HBlock 日志采集结束时间 2 小时之前的时间点。</p> <p>注意：</p> <ul style="list-style-type: none"> <li>● HBlock 日志采集的起始时间必须早于 HBlocks 日志采集的结束时间。</li> <li>● 如果 HBlock 日志采集的起始时间早于 HBlock 初始化时间，则 HBlock 初始化时间为日志采集起始时间。</li> </ul>	否

	<ul style="list-style-type: none"> <li>● 如果修改系统时间，可能导致日志文件的最后修改时间出现波动甚至错误。</li> </ul>	
endTime	<p>HBlock 日志采集的结束时间。</p> <p>类型：时间戳</p> <p>取值：unix 时间戳（UTC），精确到毫秒。默认值为当前时间。</p> <p>注意：</p> <ul style="list-style-type: none"> <li>● HBlock 日志采集的起始时间必须早于 HBlocks 日志采集的结束时间。</li> <li>● HBlock 日志采集的结束时间必须晚于 HBlock 初始化时间。</li> <li>● 日志文件的最后修改时间大于结束时间时，如果存在多个大于结束时间的同类型日志，则 HBlock 日志采集时间值最小的那个日志文件。</li> </ul>	否
servers	<p>要采集 HBlock 日志的服务器 ID。</p> <p>类型：数组</p> <p>取值：可以填写多个服务器 ID，以英文逗号（,）隔开。</p> <p>默认采集所有服务器。</p>	否
logTypes	<p>采集的日志类型</p> <p>类型：数组</p> <p>取值：</p> <ul style="list-style-type: none"> <li>● Config: 配置相关的日志。</li> <li>● System: 系统相关的日志。</li> <li>● Data: 数据处理相关的日志（仅集群版支持）。</li> <li>● Coordination: 内部协调服务相关的日志（仅集群版支持）。</li> </ul> <p>默认采集所有类型的日志。</p>	否

outputDirectory	HBlock 日志采集后存放的目录，为绝对路径。 类型：字符串 取值：如果不指定，默认存放在被请求服务器的 HBlock 安装目录下。以 collected_logs/hblock_logs_id_yyyyMMddHHmss_yyyyMMddHHmss.zip 命名，其中： <ul style="list-style-type: none"> <li>● id：本次日志请求的唯一标识符。</li> <li>● yyyyMMddHHmss：日志采集的起始时间和结束时间，UTC+0 时间。</li> </ul>	否
-----------------	---	---

● 响应结果

名称	描述
id	日志 ID。

● 请求示例 1

采集服务器 ID 为 hblock\_1 和 hblock\_2、时间为 2022-08-07 10:00:00 到 2022-08-07 22:00:00、配置和系统的相关的日志，并将日志文件存放在 /mnt/storage01 下。

```

POST /rest/v1/system/logcollect HTTP/1.1
Date: Mon, 08 Aug 2022 05:57:49 GMT
Authorization: HBlock userName:signature
Content-Length: 192
Content-Type: text/plain
Host:192.168.0.121:1443

{
  "startTime": "1659837600000",
  "endTime": "1659880800000",
  "servers": ["hblock_1","hblock_2"],
  "logTypes": ["Config","System"],
  "outputDirectory": "/mnt/storage01"
    
```

```
}
```

- 响应示例 1

```
HTTP/1.1 202 Accepted
x-hblock-request-id: 68a53a147d504d5087841192adef799e
Connection: keep-alive
Content-Length: 50
Date: Mon, 8 Aug 2022 05:57:52 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "id": "7889764ce31444d2b06f8c2b8201c836"
  }
}
```

- 请求示例 2

采集 HBlock 最近两个小时所有服务器上所有类型的日志，并将日志文件存放在 /mnt/storage01 下。

```
POST /rest/v1/system/logcollect HTTP/1.1
Date: Mon, 08 Aug 2022 06:18:12 GMT
Authorization: HBlock userName:signature
Content-Length: 45
Content-Type: text/plain
Host:192.168.0.121:1443

{
  "outputDirectory": "/mnt/storage01"
}
```

- 响应示例 2

```
HTTP/1.1 202 Accepted
```

```
x-hblock-request-id: f9cbe1e2e7564f00becc8903aaa7c9e6
```

```
Connection: keep-alive
```

```
Content-Length: 50
```

```
Date: Mon, 8 Aug 2022 06:18:14 GMT
```

```
Content-Type: application/json;charset=utf-8
```

```
Server: HBlock
```

```
{  
  "data": {  
    "id": "8c042a768cc74b15b74669d7ebec683c"  
  }  
}
```

#### 4.11.4 批量查询采集的日志

此操作用来批量查询采集的日志。

##### ● 请求语法

```
GET /rest/v1/system/logcollect?filter=filter HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

##### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为： <ul style="list-style-type: none"> <li>● status: 日志采集的状态                             <ul style="list-style-type: none"> <li>■ Processing: 日志采集中。</li> <li>■ Succeeded: 采集成功。</li> <li>■ PartiallySucceeded: 部分采集成功。</li> <li>■ Failed: 采集失败。</li> </ul> </li> <li>● logTypes: 采集日志的类型                             <ul style="list-style-type: none"> <li>■ Config: 配置相关的日志。</li> <li>■ System: 系统相关的日志。</li> <li>■ Data: 数据处理相关的日志（仅集群版支持）。</li> <li>■ Coordination: 内部协调服务相关的日志（仅集群版支持）。</li> </ul> </li> </ul>	否

##### ● 响应结果

名称	描述
----	----

logs	日志列表信息，包含：id、startTime、endTime、servers、logTypes、file、status、size、createTime。 类型：数组
id	日志 ID。
startTime	HBlock 日志采集的起始时间。
endTime	HBlock 日志采集的结束时间。
servers	HBlock 日志采集的服务器 ID。
logTypes	采集的 HBlock 日志类型： <ul style="list-style-type: none"> <li>● <b>Config</b>: 配置相关的日志。</li> <li>● <b>System</b>: 系统相关的日志。</li> <li>● <b>Data</b>: 数据处理相关的日志。</li> <li>● <b>Coordination</b>: 内部协调服务相关的日志（仅集群版支持）。</li> </ul>
file	日志文件存储路径及名称。
status	日志采集的状态： <ul style="list-style-type: none"> <li>● <b>Processing</b>: 日志采集中。</li> <li>● <b>Succeeded</b>: 采集成功。</li> <li>● <b>PartiallySucceeded</b>: 部分采集成功。</li> <li>● <b>Failed</b>: 采集失败。</li> </ul>
size	采集到的日志文件大小，单位为字节。
createTime	日志采集请求的创建时间。

● 请求示例 1

查询所有的采集日志。

```
GET /rest/v1/system/logcollect HTTP/1.1
Date: Mon, 05 Sep 2022 07:33:24 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

## ● 响应示例 1

```
HTTP/1.1 200 OK
x-hblock-request-id: ae3b8ec3785e4792bf8aeddd1ed3ab26
Connection: keep-alive
Content-Length: 1994
Date: Mon, 05 Sep 2022 07:33:26 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "logs": [
      {
        "id": "940a91055c9a4890930924a7fbb5564b",
        "startTime": 1662354629144,
        "endTime": 1662361829144,
        "servers": [
          "hblock_1",
          "hblock_2",
          "hblock_3"
        ],
        "logTypes": [
          "Data",
          "Coordination"
        ],
        "file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_940a910
55c9a4890930924a7fbb5564b_20220905051029_20220905071029.zip",
        "status": "Succeeded",
        "size": 271859170,
        "createTime": 1662361843900
      },
      {
        "id": "a334d6cddc3d40aebaa64805716ba371",
        "startTime": 1662103788274,
```

```
    "endTime": 1662110988274,
    "servers": [
      "hblock_1",
      "hblock_2",
      "hblock_3"
    ],
    "logTypes": [
      "System",
      "Data"
    ],
    "file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_a334d6c
ddc3d40aebaa64805716ba371_20220902072948_20220902092948.zip",
      "status": "Succeeded",
      "size": 380924434,
      "createTime": 1662110999389
    },
    {
      "id": "072b268df10647a495f020270caf9c64",
      "startTime": 1661739658000,
      "endTime": 1661743258183,
      "servers": [
        "hblock_1",
        "hblock_2",
        "hblock_3"
      ],
      "logTypes": [
        "Config",
        "System",
        "Data",
        "Coordination"
      ],
      "file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_072b268
df10647a495f020270caf9c64_20220829022058_20220829032058.zip",
```

```
    "status": "Succeeded",
    "size": 317186892,
    "createTime": 1661743266683
  },
  {
    "id": "e57dc4e251644eb2b272b7e4e6e7a1a9",
    "startTime": 1661735959011,
    "endTime": 1661743159011,
    "servers": [
      "hblock_1",
      "hblock_2",
      "hblock_3"
    ],
    "logTypes": [
      "Data"
    ],
    "file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_e57dc4e
251644eb2b272b7e4e6e7a1a9_20220829011919_20220829031919.zip",
    "status": "Succeeded",
    "size": 199552542,
    "createTime": 1661743174470
  },
  {
    "id": "d9862657b0744878b1c6ad2e8d9ecb88",
    "startTime": 1661735856947,
    "endTime": 1661743056947,
    "servers": [
      "hblock_1",
      "hblock_2",
      "hblock_3"
    ],
    "logTypes": [
      "Config",
      "System",
```

```
        "Data",
        "Coordination"
    ],
    "file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_d986265
7b0744878b1c6ad2e8d9ecb88_20220829011736_20220829031736.zip",
        "status": "Succeeded",
        "size": 339722784,
        "createTime": 1661743058833
    }
]
}
}
```

### ● 请求示例 2

查询日志类型为 Config 的日志。

```
GET /rest/v1/system/logcollect?filter=logTypes:Config HTTP/1.1
Date: Mon, 5 Sep 2022 07:38:08 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例 2

```
HTTP/1.1 200 OK
x-hblock-request-id: 3906705dd9e946ffb640c1b97649bee8
Connection: keep-alive
Content-Length: 839
Date: Mon, 5 Sep 2022 07:38:10 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "logs": [
      {
```

```
"id": "072b268df10647a495f020270caf9c64",
"startTime": 1661739658000,
"endTime": 1661743258183,
"servers": [
  "hblock_1",
  "hblock_2",
  "hblock_3"
],
"logTypes": [
  "Config",
  "System",
  "Data",
  "Coordination"
],
"file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_072b268
df10647a495f020270caf9c64_20220829022058_20220829032058.zip",
"status": "Succeeded",
"size": 317186892,
"createTime": 1661743266683
},
{
  "id": "d9862657b0744878b1c6ad2e8d9ecb88",
  "startTime": 1661735856947,
  "endTime": 1661743056947,
  "servers": [
    "hblock_1",
    "hblock_2",
    "hblock_3"
  ],
  "logTypes": [
    "Config",
    "System",
    "Data",
    "Coordination"
```

```
    ],  
    "file":  
    "192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_d986265  
7b0744878b1c6ad2e8d9ecb88_20220829011736_20220829031736.zip",  
    "status": "Succeeded",  
    "size": 339722784,  
    "createTime": 1661743058833  
  }  
]  
}  
}
```

### 4.11.5 查询单个采集的日志

此操作用来查询单个采集的日志。

#### ● 请求语法

```
GET /rest/v1/system/logcollect/id HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

#### ● 请求参数

参数	描述	是否必须
id	日志 ID。 如果不填写，查询所有的采集日志。	否

#### ● 响应结果

名称	描述
logs	日志列表信息，包含：id、startTime、endTime、servers、logTypes、file、status、size、createTime。 类型：数组
id	日志 ID。
startTime	HBlock 日志采集的起始时间。
endTime	HBlock 日志采集的结束时间。
servers	日志采集的服务器 ID。
logTypes	采集的日志类型： <ul style="list-style-type: none"> <li>● Config: 配置相关的日志。</li> <li>● System: 系统相关的日志。</li> <li>● Data: 数据处理相关的日志（仅集群版支持）。</li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Coordination:</b> 内部协调服务相关的日志（仅集群版支持）。</li> </ul>
<b>file</b>	日志文件存储路径及名称。
<b>status</b>	日志采集的状态： <ul style="list-style-type: none"> <li>● <b>Processing:</b> 日志采集中。</li> <li>● <b>Succeeded:</b> 采集成功。</li> <li>● <b>PartiallySucceeded:</b> 部分采集成功。</li> <li>● <b>Failed:</b> 采集失败。</li> </ul>
<b>size</b>	采集到的日志文件大小，单位为字节。
<b>createTime</b>	日志采集请求的创建时间。

### ● 请求示例

查询 ID 为 5f84a6ea8edf4f9ea1debe601ce15c22 的日志。

```
GET /rest/v1/system/logcollect/5f84a6ea8edf4f9ea1debe601ce15c22 HTTP/1.1
Date: Mon, 5 Sep 2022 07:47:06 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 83e8981108e948719fb64f7a977fede5
Connection: keep-alive
Content-Length: 418
Date: Mon, 5 Sep 2022 07:47:06 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "id": "8b5615b7f56a4f1196fafd113a1a0f18",
    "startTime": 1662363858000,
    "endTime": 1662363918504,
```

```
"servers": [
  "hblock_1",
  "hblock_2",
  "hblock_3"
],
"logTypes": [
  "Config",
  "System",
  "Data",
  "Coordination"
],
"file":
"192.168.0.72:/mnt/storage01/CTYUN_HBlock_Plus_3.5.0/collected_logs/hblock_logs_8b5615b
7f56a4f1196fafd113a1a0f18_20220905074418_20220905074518.zip",
  "status": "Succeeded",
  "size": 319185251,
  "createTime": 1662363939344
}
}
```

## 4.11.6 下载采集的日志文件

此操作用来下载采集的日志文件。

**注意：**只有采集成功的日志和部分采集成功的日志才能被下载。

### ● 请求语法

```
GET /rest/v1/system/logcollect/id/file HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

### ● 请求参数

参数	描述	是否必须
id	日志 ID。	是

### ● 请求示例

下载 ID 为 903fec5908534445b847a2b13bd205e2 的日志。

```
GET /rest/v1/system/logcollect/903fec5908534445b847a2b13bd205e2/file HTTP/1.1
```

Date: Mon, 05 Sep 2022 08:26:41 GMT

Authorization: HBlock *userName:signature*

Host:192.168.0.121:1443

### ● 响应示例

```
HTTP/1.1 200 OK
```

x-hblock-request-id: c0958bc7287c4ca4a515e9ac5e32863d

Connection: keep-alive

Content-Length: 28950

Date: Mon, 5 Sep 2022 08:26:42 GMT

Content-Disposition:

attachment;filename=hblock\_logs\_903fec5908534445b847a2b13bd205e2\_20220905082042\_2022090

```
5082242.zip  
Content-Type: application/zip;charset=utf-8  
Server: HBlock  
  
{ ..... }
```

## 4.11.7 删除指定的采集日志

此操作用来删除指定的采集日志。

- 请求语法

```
DELETE /rest/v1/system/logcollect/id HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

- 请求参数

参数	描述	是否必须
id	日志 ID。 如果不填写，删除所有采集日志。	否

- 请求示例

删除 ID 为 5f84a6ea8edf4f9ea1debe601ce15c22 的日志。

```
DELETE /rest/v1/system/logcollect/5f84a6ea8edf4f9ea1debe601ce15c22 HTTP/1.1
Date: Mon, 08 Aug 2022 08:55:18 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 9bed087398184a77b7ef63eabb0670d3
Connection: keep-alive
Date: Mon, 8 Aug 2022 08:55:20 GMT
Server: HBlock
```

## 4.11.8 删除所有的采集日志

此操作用来删除所有的采集日志。

- 请求语法

```
DELETE /rest/v1/system/logcollect HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

- 请求示例

删除所有的采集日志。

```
DELETE /rest/v1/system/logcollect HTTP/1.1  
Date: Mon, 08 Aug 2022 09:01:57 GMT  
Authorization: HBlock userName:signature  
Host:192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: d5e2973621064ed7b2e846b3874535d3  
Connection: keep-alive  
Date: Mon, 8 Aug 2022 09:02:00 GMT  
Server: HBlock
```

### 4.11.9 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接服务器。
400	ExceedMaxValue	Value <i>value</i> at 'number' failed to satisfy constraint: Argument must be less than or equal to <i>value</i> .	超过最大值。
400	ExceedThreshold	The number of <i>operation</i> cannot exceed value.	超出了系统允许的最大个数，无法执行该操作。
400	InsufficientServerSpace	This method is not allowed because the free space of <i>directory</i> must be greater than or equal to <i>value</i> for server <i>serverId[, serverId...]</i> .	存储空间不足。
400	InvalidDirectory	The <i>argument</i> must be a directory specified with an absolute path.	输出路径必须是带有绝对路径的目录。
400	InvalidEnumValue	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must satisfy enum value set: [ <i>value1, value2... </i> ]	枚举参数不合法。
400	InvalidPositiveInteger	Value <i>value</i> at ' <i>number</i> ' failed to	取值必须为正整数。

		satisfy constraint: Argument must be positive integer.	
400	InvalidTime	The end time must be later than the start time.	结束时间必须晚于开始时间。
400	InvalidTimeFormat	Value <i>value</i> at ' <i>argument</i> ' failed to satisfy constraint: Argument must conform to <i>Format</i> format.	时间格式不合法。
400	StandaloneModeNotAllowed	' <i>operation</i> ' is not supported by standalone mode of HBlock.	单机版本 HBlock 不支持该操作。
400	TooLargeFile	The file is too large. It is recommended to narrow the query range.	日志文件过大，建议缩小查询范围。
403	PermissionDenied	Failed to action because user has no permission.	权限不足，无法执行该操作。
404	NoSuchLog	The log with ID <i>ID</i> does not exist.	日志文件不存在。
409	InvalidLogStatus	The status of log collection task with ID <i>ID</i> is ' <i>status</i> ', the request is invalid.	日志状态冲突，无法执行该操作。

## 4.12 HBlock 系统设置

### 4.12.1 修改管理员密码

此操作用来修改管理员密码。

- 请求语法

```
PUT /rest/v1/system/user/password HTTP/1.1
Date: date
Content-Type: application/json; charset=utf-8
Content-Length: length
Host: ip:port
Authorization: authorization

{
  "userName": userName,
  "newPassword": newPassword
}
```

- 请求参数

名称	描述	是否必须
userName	HBlock 的管理员用户名。 类型：字符串 取值：与初始化设置的用户名保持一致。	是
newPassword	用户设置的新密码的明文。 类型：字符串 取值：8~16 位字符串，包含字母、数字或特殊符号(~ ! @ # \$ % ^ & * ( ) _ + [ ] { }   ; : , . / < > ?)。	是

- 请求示例

修改管理员密码。

```
PUT /rest/v1/system/user/password HTTP/1.1
Date: Wed, 24 Jan 2024 07:54:58 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 67
Host:192.168.0.110:1443
```

```
{
  "userName": "storuser",
  "newPassword": "*****"
}
```

- 响应示例

```
HTTP/1.1 204 No Content
x-hblock-request-id: 914ac22b6fb74f5ba5dc085e35718294
Connection: keep-alive
Date: Wed, 24 Jan 2024 07:54:58 GMT
Server: HBlock
```

## 4.12.2 邮件设置

### 4.12.2.1 设置邮件

此操作用来设置或者修改邮件配置信息。

**注意：** 如果邮件服务器地址是 IPv6 地址，为了确保邮件能发送成功，建议每台 HBlock 服务器上都有一个能连接到邮件服务器的 IPv6 地址。。

#### ● 请求语法

```
PUT /rest/v1/system/config/notification HTTP/1.1
```

```
Date: date
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: length
```

```
Host: ip:port
```

```
Authorization: authorization
```

```
{
  "status": status,
  "smtpHost": smtpHost,
  "smtpPort": smtpPort,
  "SSL": ssl,
  "senderEmail": senderEmail,
  "password": password,
  "receiverEmail": receiverEmail
}
```

#### ● 请求参数

参数	描述	是否必须
status	是否启用邮件通知功能。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> </ul>	是

	<ul style="list-style-type: none"> <li>● <b>Disabled:</b> 禁用。</li> </ul>	
smtpHost	SMTP 服务器。 取值: SMTP 服务器域名或 IP。 <b>注意:</b> 如果启用邮件通知时, 未在服务端保存过邮件配置, 此项必填。	否
smtpPort	SMTP 端口号。 类型: 整型 取值: [1,65535]。 <b>说明:</b> <ul style="list-style-type: none"> <li>● 如果启用了 SSL, 默认端口为 465。</li> <li>● 如果禁用 SSL, 默认端口为 25。</li> </ul>	否
SSL	是否启用 SSL。 类型: 枚举 取值: <ul style="list-style-type: none"> <li>● <b>Enabled:</b> 启用 SSL。</li> <li>● <b>Disabled:</b> 禁用 SSL。</li> </ul> 默认值为 Disabled。	否
senderEmail	发件箱。 邮箱格式 <i>local-part@domain</i> : <ul style="list-style-type: none"> <li>● <b>local-part:</b> 字符串形式, 长度 1~64, 可包含字母、数字、特殊字符 (!#\$% &amp;*+ - / = ? ^ _ ` {   } ~ .), 字母区分大小写。句点 (.) 不能作为首尾字符, 也不能连续出现。</li> <li>● <b>domain:</b> 以句点 (.) 分隔的字符串形式, 长度 1~255。通过句点 (.) 分隔开的每个字符串需要满足如下要求:                             <ul style="list-style-type: none"> <li>■ 长度 1~63。</li> <li>■ 可包含字母、数字、短横线 (-), 字母区分大小写。</li> </ul> </li> </ul>	否

	<ul style="list-style-type: none"> <li>■ 顶级域名不能是纯数字。</li> <li>■ 短横线 (-) 不能作为首尾字符。</li> </ul> <p><b>注意：</b>如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。</p>	
password	<p>邮箱授权码。</p> <p><b>说明：</b>授权码是邮箱推出的，用于第三方客户端登录的专用密码。</p> <p><b>注意：</b>如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。</p>	否
receiverEmail	<p>收件箱。可以填写多个收件箱。</p> <p>邮箱格式 <i>Local-part@domain</i>:</p> <ul style="list-style-type: none"> <li>● <i>Local-part</i>: 字符串形式，长度 1~64，可包含字母、数字、特殊字符 (!#\$% &amp;*+ - / = ? ^ _ ` {   } ~ .)，字母区分大小写。句点 (.) 不能作为首尾字符，也不能连续出现。</li> <li>● <i>domain</i>: 以句点 (.) 分隔的字符串形式，长度 1~255。通过句点 (.) 分隔开的每个字符串需要满足如下要求：                     <ul style="list-style-type: none"> <li>■ 长度 1~63。</li> <li>■ 可包含字母、数字、短横线 (-)，字母区分大小写。</li> <li>■ 顶级域名不能是纯数字。</li> <li>■ 短横线 (-) 不能作为首尾字符。</li> </ul> </li> </ul> <p><b>注意：</b>如果启用邮件通知时，未在服务端保存过邮件配置，此项必填。</p>	否

### ● 请求示例 1

设置邮件配置信息，并启用邮件通知功能。

```
PUT /rest/v1/system/config/notification HTTP/1.1
```

```
Date: Mon, 1 Aug 2022 01:53:36 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 243
Host: 192.168.0.121:1443

{
  "status": "Enabled",
  "smtpHost": "smtp.chinatelecom.cn",
  "smtpPort": 475,
  "SSL": "Enabled",
  "senderEmail": "account1@chinatelecom.cn",
  "password": "*****",
  "receiverEmail": "account2@chinatelecom.cn"
}
```

#### ● 响应示例 1

```
HTTP/1.1 204 No Content
x-hblock-request-id: 24c17819e3ed4c648f37d13f9af05537
Connection: keep-alive
Date: Mon, 1 Aug 2022 01:53:36 GMT
Server: HBlock
```

#### ● 请求示例 2

关闭邮件通知功能，但是不删除已经配置的 smtpHost 等信息。

```
PUT /rest/v1/system/config/notification HTTP/1.1
Date: Mon, 01 Aug 2022 01:56:05 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 30
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature

{
  "status": "Disabled"
}
```

```
}
```

- 响应示例 2

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 9022434a208a4c2ebf89da97f3f72976  
Connection: keep-alive  
Date: Mon, 01 Aug 2022 01:56:05 GMT  
Server: HBlock
```

#### 4.12.2.2 发送测试邮件

此操作用来发送测试邮件。

- 请求语法

```
POST /rest/v1/system/config/notification/testEmail HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

- 请求示例

发送测试邮件。

```
POST /rest/v1/system/config/notification/testEmail HTTP/1.1
Date: Thu, 17 Mar 2022 05:39:56 GMT
Content-Type: application/json; charset=utf-8
Authorization: HBlock userName:signature
Content-Length: 0
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 3de55df515314e4cb9b7b61ade66937f
Connection: keep-alive
Date: Thu, 17 Mar 2022 05:40:01 GMT
Content-Length: 0
Server: HBlock
```

### 4.12.2.3 查询邮件配置信息

此操作用来查询邮件配置信息。

#### ● 请求语法

```
GET /rest/v1/system/config/notification HTTP/1.1
```

Date: *date*

Host: *ip:port*

Authorization: *authorization*

#### ● 响应结果

名称	描述
status	是否启用邮件通知功能： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>
smtpHost	SMTP 服务器。
smtpPort	SMTP 服务器端口号。
SSL	是否启用 SSL： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>
senderEmail	发件箱。
receiverEmail	收件箱。

#### ● 请求示例

查询邮件配置信息。

```
GET /rest/v1/system/config/notification HTTP/1.1
```

Date: Thu, 17 Mar 2022 05:46:57 GMT

Authorization: HBlock *userName:signature*

Host: 192.168.0.121:1443

- 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: 4b5e0623e73a4dcd8754fd2ed504b9c1
Connection: keep-alive
Content-Length: 199
Date: Thu, 17 Mar 2022 05:46:57 GMT
Content-Type: application/json;charset=utf-8
Server: HBlock

{
  "data": {
    "status": "Disabled",
    "smtpHost": "smtp.chinatelecom.cn",
    "smtpPort": 475,
    "SSL": "Enabled",
    "senderEmail": "account1@chinatelecom.cn",
    "receiverEmail": "account2@chinatelecom.cn"
  }
}
```

#### 4.12.2.4 删除邮件配置

此操作用来删除邮件配置。

- 请求语法

```
DELETE /rest/v1/system/config/notification HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

- 请求示例

删除邮件配置。

```
DELETE /rest/v1/system/config/notification HTTP/1.1  
Date: Thu, 17 Mar 2022 05:56:29 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: 9639c0c89dca4fcb83719458a50452ac  
Connection: keep-alive  
Date: Thu, 17 Mar 2022 05:56:29 GMT  
Server: HBlock
```

## 4.12.2.5 错误码列表

HTTP status	错误码	错误信息	描述
400	InvalidReceiverEmail	Value <i>value</i> at 'receiver email' failed to satisfy constraint: Argument must conform to email format, and be separated by comma if there are multiple receivers.	接收者邮箱不合法。
400	InvalidSenderEmail	Value <i>value</i> at 'sender email' failed to satisfy constraint: Argument must conform to email format, and only one sender can be set.	发送者邮箱不合法。
400	MissingReceiverEmail	Value null at 'receiver email' failed to satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，receiverEmail 不能为空。
400	MissingSenderEmail	Value null at 'sender email' failed to satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，senderEmail 不能为空。
400	MissingSenderPassword	Value null at 'password' failed to satisfy constraint: Argument must not be null when enable email configuration.	启用邮件设置时，password 不能为空。
400	MissingSMTPHost	Value null at 'smtp host' failed to	启用邮件设置

		satisfy constraint: Argument must not be null when enable email configuration.	时，SMTP 服务器不能为空。
400	SendEmailFailed	Test email failed to send out.	发送测试邮件失败。
404	NoSuchNotificationConfiguration	Notification is not configured.	没有配置邮件通知功能。

## 4.12.3 远程协助

### 4.12.3.1 设置远程协助

此操作用来设置远程协助。

- 请求语法

```
PUT /rest/v1/system/config/remoteAccess HTTP/1.1
```

```
Date: date
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: length
```

```
Host: ip:port
```

```
Authorization: authorization
```

```
{
  "serverId": serverId,
  "status": status,
  "host": host,
  "port": port
}
```

- 请求参数

名称	描述	是否必须
serverId	服务器 ID。 如果不指定，默认设置当前被请求的服务器。	否
status	是否启用远程协助。 类型：枚举 取值： <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>	是
host	远程协助的服务端 IP 或域名。	否

	<p><b>说明：</b> 请联系软件供应商获取 HOST 信息。</p> <p>除下列情况 host 必填外，其他情况选填：</p> <ul style="list-style-type: none"> <li>● 如果启用远程协助时，此项在服务端没有保存过。</li> <li>● 如果设置远程协助时，输入了具体的端口号。</li> </ul>	
port	<p>远程协助的端口。</p> <p><b>说明：</b> 请联系软件供应商获取端口号。</p> <p>类型：整型</p> <p>取值： [1,65535]，默认 18100。</p>	否

### ● 响应结果

名称	描述
code	远程协助码，6 位数字。
host	远程协助的服务端 IP。
port	远程协助端口。

### ● 请求示例

设置远程协助。

```

PUT /rest/v1/system/config/remoteAccess HTTP/1.1
Date: Thu, 17 Mar 2022 09:20:16 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 113
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature

{
  "serverId": "hblock_1",
  "status": "Enabled",
  "host": "182.151.21.174",
  "port": 18100
}
    
```

- 响应示例

```
HTTP/1.1 200 OK
Date: Thu, 17 Mar 2022 09:20:16 GMT
Content-Length: 63
Connection:keep-alive
Content-Type: application/json; charset=utf-8
x-hblock-request-id: a839f9f8e76a4f18969f7e2637c957eb
Server: HBlock

{
  "data": {
    "code": "999342",
    "host": "182.151.21.174",
    "port": 18100
  }
}
```

### 4.12.3.2 查询远程协助

此操作用来查询服务器的远程协助配置信息。

注意：

- 如果 HBlock 还没有初始化，那么只显示本机的信息。
- 仅支持查询有远程协助配置的服务器。

#### ● 请求语法

```
GET /rest/v1/system/config/remoteAccess?filter=filter&range=i-j HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

#### ● 请求参数

参数	描述	是否必须
filter	查询的过滤条件。 支持的查询条件为：serverId（服务器 ID）。	否
range	查询服务器的范围，格式为 i-j，i 和 j 为正整数，0<j-i<1000。按照服务器 ID 进行排序，返回第 i 到第 j 的服务器。 如果输入的查询范围超过服务器的总个数，返回的结果为空，如果未指定 range，则返回所有符合条件的服务器。	否

#### ● 响应结果

名称	描述
serverId	服务器 ID，若还没有初始化，则不显示此项。
status	远程协助是否启用。 <ul style="list-style-type: none"> <li>● Enabled: 启用。</li> <li>● Disabled: 禁用。</li> </ul>
host	远程协助的服务器 Host。

port	远程协助的端口。
code	远程协助码。

- 请求示例

查询远程协助。

```
GET /rest/v1/system/config/remoteAccess HTTP/1.1
Date: Thu, 17 Mar 2022 09:34:04 GMT
Host: 192.168.0.121:1443
Authorization: HBlock userName:signature
```

- 响应示例

```
HTTP/1.1 200 OK
Date: Thu, 17 Mar 2022 09:34:04 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 214
Connection: keep-alive
x-hblock-request-id: 0cda25588df44123ae5113876bd91ca0
Server: HBlock

{
  "data": {
    "servers": [
      {
        "serverId": "hblock_1",
        "status": "Enabled",
        "host": "182.151.21.174",
        "port": 18100,
        "code": "999342"
      },
      {
        "serverId": "hblock_2",
        "status": "Enabled",
        "host": "182.151.21.174",
        "port": 18100,
```

```
    "code": "258699"  
  }  
]  
}  
}
```

### 4.12.3.3 删除远程协助

此操作用来删除远程协助配置。

说明：如果不填写 `serverId`，表示删除当前被请求的服务器的远程协助配置。

- 请求语法

```
DELETE /rest/v1/system/config/remoteAccess/serverId HTTP/1.1  
Date: date  
Host: ip:port  
Authorization: authorization
```

- 请求示例

删除远程协助配置。

```
DELETE /rest/v1/system/config/remoteAccess/hblock_1 HTTP/1.1  
Date: Thu, 17 Mar 2022 09:37:30 GMT  
Authorization: HBlock userName:signature  
Host: 192.168.0.121:1443
```

- 响应示例

```
HTTP/1.1 204 No Content  
x-hblock-request-id: a4e3bc5f38a24992ab70da684388d6f6  
Connection: keep-alive  
Date: Thu, 17 Mar 2022 09:37:30 GMT  
Server: HBlock
```

#### 4.12.3.4 错误码列表

HTTP status	错误码	错误信息	描述
400	CanNotConnectToServer	Can not connect to the server <i>IP[:port]</i> .	无法连接指定的服务器。
400	InvalidHost	The host is invalid.	host 无效。
400	MissingRemoteAccessHost	Value null at 'host' failed to satisfy constraint: Argument must not be null.	启用远程协助时，host 不能为空。
404	NoSuchRemoteAccessConfiguration	Remote access is not configured.	没有配置远程协助。
409	RemoteAccessAlreadyEnabled	Remote access to <i>Host:Port</i> has been enabled, and the remote access code is: <i>CODE</i> . If you want to change the Host and Port, please disable the remote access first.	已经启用了远程协助，再次启用时 Host、Port 和第一次填写的不一致。

## 4.13 还原 HBlock 集群中的服务器（集群版适用）

### 4.13.1 还原 HBlock 集群中的服务器

此操作用来将集群中指定服务器的 HBlock 配置信息，还原至目标服务器。

注意：

- 非基础节点服务器损坏无法启动的时候，可以通过其他服务器移除该损坏的服务器，也可以还原该服务器；但基础节点服务器损坏，只能还原服务器。
- 指定服务器上的所有 HBlock 服务已停止。
- 目标服务器的 IP、端口、HBlock 版本和指定服务器的 IP、端口、HBlock 版本必须一致。
- 需要先在目标服务器上执行 **install** 命令，再在集群中其他可用服务器上执行此操作。

#### ● 请求语法

```
PUT /rest/v1/system/recover/sourceServerId HTTP/1.1
```

```
Date: date
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: length
```

```
Host: ip:port
```

```
Authorization: authorization
```

```
{
  "destServer":{
    "diskPaths": [
      {"path": path,
       "capacityQuota": capacityvalue
      },
      {"path": path,
       "capacityQuota": capacityvalue
      },
      .....
    ]
  }
}
```

```

    ]
  }
}
    
```

### ● 请求参数

参数	描述	是否必须
sourceServerId	源服务器 ID。	是
diskPaths	目标服务器的数据目录属性集合。包括 path、capacityQuota。 类型：数组。	是
path	指定数据目录，支持多个数据目录，至少指定一个数据目录。数据目录用于存储数据，建议不要与操作系统共用磁盘或文件系统。 <b>注意：</b> 数据目录中不能有逗号（,）。 类型：字符串列表	是
capacityQuota	指定数据目录的容量配额，即针对加入到服务器中的每个数据目录，HBlock 可写入的数据总量。当 HBlock 的使用空间一旦达到配额，就立刻阻止数据写入，不允许再使用超出配额的空间。 类型：长整型。 取值：小于数据目录所在磁盘的总容量，单位是字节。负整数表示无限制写入，0 表示禁止写入。默认不限制写入。 <b>注意：</b> 如果相同的数据目录出现多次，以第一次出现的数据目录的容量配额为准。	否

### ● 请求示例

将服务器 ID 为 hblock\_10 的 HBlock 服务器的配置信息还原至目标服务器。

```
PUT /rest/v1/system/recover/hblock_10 HTTP/1.1
```

```
Date: Wed, 11 Oct 2023 07:13:35 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 82
Host: 192.168.0.72:1443
Authorization:HBlock userName:signature
```

```
{
  "destServer":{
    "diskPaths":[
      {"path":"/mnt/storage01"},
      {"path":"/mnt/storage02"}
    ]
  }
}
```

- 响应示例

```
HTTP/1.1 204 No Content
Date: Wed, 11 Oct 2023 07:13:35 GMT
Content-Type: application/json; charset=utf-8
Connection: keep-alive
x-hblock-request-id: 280f00aeb5064f73ab1ce379f28ea789
Server: HBlock
```

### 4.13.2 错误码列表

HTTP status	错误码	错误信息	描述
400	InvalidDestServer	Failed to recover because the IPs, ports, product type and version must be same with old server.	新旧服务器 IP、Port、产品类型和版本号需要保持一致。
400	InvalidDestServerStatus	The server to be operated is not available.	操作失败，服务器不可用。
409	PortConflict	The following port is/ports are in use. <i>server_ID/IP/local server: port_name port[, port_name port...]</i>	端口冲突。

## 4.14 查询 HBlock 版本

此操作用来查询 HBlock 版本信息。

### ● 请求语法

```
GET /rest/v1/system/version HTTP/1.1
Date: date
Host: ip:port
Authorization: authorization
```

### ● 响应结果

名称	描述
version	HBlock 的版本号。
type	HBlock 的版本类型： <ul style="list-style-type: none"><li>● Stand-alone: 单机版。</li><li>● Cluster: 集群版本。</li></ul>

### ● 请求示例

查询 HBlock 的版本信息。

```
GET /rest/v1/system/version HTTP/1.1
Date: Thu, 17 Mar 2022 08:00:22 GMT
Authorization: HBlock userName:signature
Host:192.168.0.121:1443
```

### ● 响应示例

```
HTTP/1.1 200 OK
x-hblock-request-id: ab2219e1469849fb81742a9395b9f22a
Connection: keep-alive
Content-Length: 45
Date: Thu, 17 Mar 2022 08:00:22 GMT
Content-Type: application/json;charset=utf-8
```

Server: HBlock

```
{  
  "data": {  
    "version": "3.5.0",  
    "type": "Cluster"  
  }  
}
```

## 5 附录

---

### 5.1 转义字符表

对于下列特殊字符，在响应体中会进行转义。

转义前	转义后
<	&lt;
>	&gt;
(	&#40;
)	&#41;
&	&amp;

## 5.2 HBlock 服务

服务	服务名称	作用
stor:mdm	元数据管理服务	管理整个系统的元数据
stor:fc	故障转移控制服务	进行系统健康检测，实现故障转移控制
stor:ls	日志服务	提供基于日志的数据同步功能
stor:ds	数据服务	管理用户的文件数据块
stor:cs	协调服务	监视各服务器的状态，触发通知事件，确保集群服务高可用
stor:ms	管理服务	处理请求信息，维护集群运行状态。
stor:ws	监控服务	监控各个服务的状态，并负责服务的启动
stor:ps	协议解析服务	负责 iscsi 协议解析与数据存储
stor:ag	数据采集服务	负责采集性能数据。
socat	socat	在打开进行远程协助时，用于集群内各服务器之间的数据通信
stor:ua	升级监听服务	负责接收升级请求，执行升级相关操作。

## 5.3 用户事件列表

### ● 服务器

事件	描述
AddServer	添加服务器
RemoveServer	移除服务器
SetServer	设置服务器属性
DeleteTargetPortalIP	删除服务器 Target 门户 IP
RestartServer	重启服务器
RecoverServer	还原服务器
AddPath	添加数据目录
RemovePath	移除数据目录
StartServer	启动服务
StopServer	停止服务
SetPath	修改数据目录

### ● iSCSI Target

事件	描述
CreateTarget	创建 Target
DeleteTarget	删除 Target
SetTarget	设置 Target 属性
MigrateTarget	迁移 Target
DeleteCHAP	删除 CHAP
DeleteConnection	删除连接

## ● 卷

事件	描述
CreateLUN	创建卷
DeleteLUN	删除卷
SetLUN	设置卷属性
ExpandLUN	扩容卷
EnableLUN	启用卷
DisableLUN	禁用卷
SwitchLUN	卷主备切换

## ● 系统

事件	描述
Login	登录
SetMailConfig	设置邮件通知
DeleteMailConfig	删除邮件通知
SendTestMail	发送测试邮件
SetRemoteAccess	设置远程协助
DeleteRemoteAccess	删除远程协助
ImportLicense	导入软件许可证
SetPassword	设置密码
StartLogCollect	发起日志收集请求
DeleteLogCollect	删除日志收集请求
SetAlarmMuteStatus	修改告警静默状态
ManuallyResolveAlarm	手动解除告警

事件	描述
Setup	初始化

## 5.4 系统事件列表

### ● 服务器 (Server)

事件	描述
ServiceUnavailable	服务不可用
ServiceAvailable	服务可用
ServerAdded	服务器添加
ServerRemoved	服务器移除
ServerRecovered	服务器还原
CapacityQuotaUsageExceedsThreshold	配额使用率超阈值
CapacityQuotaUsageBelowThreshold	配额使用率恢复正常
CapacityQuotaUsageApproachLimit	配额用尽
CapacityQuotaUsageBelowLimit	配额低于上限
DiskUsageExceedsThreshold	磁盘使用率超阈值
DiskUsageBelowThreshold	磁盘使用率恢复正常
ProtocolServiceAbnormal	协议解析服务异常
ProtocolServiceResumed	协议解析服务恢复

### ● 数据目录 (Disk)

事件	描述
DiskIOError	磁盘 IO 错误
DiskIOResumed	磁盘 IO 恢复
DiskWriteSpeedTooSlow	磁盘写入速度慢
DiskWriteSpeedResumed	磁盘写入速度恢复正常
PathAdded	数据目录添加
PathRemoved	数据目录移除
CapacityQuotaUsageExceedsThreshold	配额使用率超阈值

CapacityQuotaUsageBelowThreshold	配额使用率恢复正常
CapacityQuotaUsageApproachLimit	配额用尽
CapacityQuotaUsageBelowLimit	配额低于上限
DiskUsageExceedsThreshold	磁盘使用率超阈值
DiskUsageBelowThreshold	磁盘使用率恢复正常
DiskPathHealthStatusWarning	数据目录健康状态警告
DiskPathHealthStatusError	数据目录健康状态错误
DiskPathHealthStatusResumed	数据目录健康状态恢复
DataServiceHealthStatusWarning	数据服务健康状态警告
DataServiceHealthStatusError	数据服务健康状态错误
DataServiceHealthStatusResumed	数据服务健康状态恢复

● 卷 (LUN)

事件	描述
ActiveStandbySwitched	卷主备切换
InsufficientFDForLUNToWrite	可用故障域数量不满足卷写入要求
SufficientFDForLUNToWrite	可用故障域数量满足卷写入要求

● 系统 (System)

事件	描述
ReachLicenseMaxCapacity	许可证容量达到上限
LicenseMaintenanceExpired	许可证过保
LicenseExpired	许可证过期
LicenseImported	许可证导入
DiskUsageExceedsThreshold	系统整体磁盘使用率超过阈值

DiskUsageBelowThreshold	系统整体磁盘使用率恢复正常
DataResumed	数据恢复
DataLowRedundancy	数据降级
DataBalanceStart	数据均衡开始
DataBalanceProgress	数据均衡过程
DataBalanceFailed	数据均衡失败
DataBalanceEnd	数据均衡结束
DataAccessFailed	数据无法访问
CapacityQuotaUsageExceedsThreshold	配额使用率超阈值
CapacityQuotaUsageBelowThreshold	配额使用率恢复正常
CapacityQuotaUsageApproachLimit	配额用尽
CapacityQuotaUsageBelowLimit	配额低于上限
DiskUsageExceedsThreshold	磁盘使用率超阈值
DiskUsageBelowThreshold	磁盘使用率恢复正常

● 故障域模块 (FaultDomain)

事件	描述
FaultDomainWarning	故障域状态变为警告
FaultDomainError	故障域状态变为错误
FaultDomainResumed	故障域状态恢复正常

## 5.5 监控指标

- 数据粒度

监控指标的粒度可以分为“精细”和“粗糙”，具体含义如下：

精细类别	数据粒度	数据保留时长	说明
精细	20 秒	2 小时	每 20 秒采集一次实时数据，生成 1 个数据点，每个数据点保留 2 小时。
	1 分钟	6 小时	基于 20s 粒度采集数据聚合 1 分钟粒度数据，保留时长 6 小时。
	5 分钟	1 天	基于 20s 粒度采集数据聚合 5 分钟粒度数据，保留时长 1 天。
	1 小时	7 天	基于 20s 粒度采集数据聚合 1 小时粒度数据，保留时长 7 天。
	1 天	1 年	基于 20s 粒度采集数据聚合 1 天粒度数据，保留时长 1 年。
粗糙	5 分钟	2 小时	基于 20s 粒度采集数据聚合 5 分钟粒度数据，保留时长 2 小时。
	1 小时	1 天	基于 20s 粒度采集数据聚合 1 小时粒度数据，保留时长 1 天。
	1 天	1 个月	基于 20s 粒度采集数据聚合 1 天粒度数据，保留时长 1 个月。
	1 周	6 个月	基于 20s 粒度采集数据聚合 1 周粒度数据，保留时长 6 个月。
	1 个月	1 年	基于 20s 粒度采集数据聚合 1 个月粒度数据，保留时长 1 年。

- 监控指标

监控对象 (dimension)	监控指标 (metric)	说明	单位	数据粒度
system	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwith	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwith	客户端从 HBlock 读取数据的带宽。	字节/s	精细
	W_Bandwith	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。系统在一个采集周期内，读写操作平均时延，反映 HBlock 处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 写入数据的时延，系统在一个采集周期内，写操作平均时延，反映 HBlock 处理写请求的时长。	ms	精细
	R_Latency	客户端从 HBlock 读取数据的时延，集群在一个采集周期内，读操作平均时延，反映 HBlock 处理读请求的时长。	ms	精细
	Path_Cap	数据目录总容量。	字节	粗糙
	Path_Used	数据目录已用容量。	字节	粗糙
Path_Rate	数据目录平均使用率，即所有数据	%	粗糙	

		目录使用率的平均值。		
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的所有目录容量配额的总和。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额平均使用率，所有数据目录 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 的平均值。	%	粗糙
server	CPU_Rate	服务器 CPU 使用率。	%	精细
	Mem_Rate	服务器内存使用率。	%	精细
	Mem_Total	服务器内存总量。	字节	精细
	Mem_Used	服务器内存使用量。	字节	精细
	IOPS	客户端与 HBlock 之间的总 IOPS。	无	精细
	R_IOPS	客户端从 HBlock 读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 写入数据的 IOPS。	无	精细
	Bandwith	客户端与 HBlock 之间的总带宽。	字节/s	精细
	R_Bandwith	客户端从 HBlock 读取数据的带宽。	字节/s	精细
	W_Bandwith	客户端向 HBlock 写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延。采集周期内，服务器关联卷的读写时延平均值。	ms	精细
	W_Latency	客户端到 HBlock 写时延，采集周期内，服务器关联卷的写时延平均	ms	精细

		值。		
	R_Latency	客户端从 HBlock 读取数据的时延，采集周期内，服务器关联卷的平均读时延。	ms	精细
	Path_Cap	服务器的数据目录总容量。	字节	粗糙
	Path_Used	服务器的数据目录已用容量。	字节	粗糙
	Path_Rate	服务器的数据目录平均使用率，即服务器上所有数据目录使用率的平均值。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的所有目录容量配额的和。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额平均使用率，即所有数据目录 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$ 的平均值。	%	粗糙
disk	Path_Cap	数据目录总容量。	字节	粗糙
	Path_Used	数据目录已用容量。	字节	粗糙
	Path_Rate	数据目录平均使用率。	%	粗糙
	Path_Cap_Quota	HBlock 可用空间大小，即用户给 HBlock 分配的容量配额。	字节	粗糙
	Path_Cap_Quota_Used	磁盘文件系统中，HBlock 数据占用的空间大小。	字节	粗糙
	Path_Cap_Quota_Rate	数据目录容量配额使用率，即 $\text{Path\_Cap\_Quota\_Used}/\text{Path\_Cap\_Quota}$	%	粗糙

		ota。		
LUN	IOPS	客户端与 HBlock 卷之间的总 IOPS	无	精细
	R_IOPS	客户端从 HBlock 卷读取数据的 IOPS。	无	精细
	W_IOPS	客户端向 HBlock 卷写入数据的 IOPS。	无	精细
	Bandwith	客户端与 HBlock 卷的之间的总带宽。	字节/s	精细
	R_Bandwith	客户端从 HBlock 卷读取数据的带宽。	字节/s	精细
	W_Bandwith	客户端向 HBlock 卷写入数据的带宽。	字节/s	精细
	Latency	客户端与 HBlock 之间的总时延，卷在一个采集周期内，读写操作平均时延，反映 HBlock 单卷处理读写请求的时长。	ms	精细
	W_Latency	客户端向 HBlock 卷写入数据的时延，卷在一个采集周期内，写操作平均时延，反映 HBlock 单卷处理写请求的时长。	ms	精细
R_Latency	客户端从 HBlock 卷读取数据的时延，卷在一个采集周期内，读操作平均时延，反映 HBlock 单卷处理读请求的时长。	ms	精细	

## 5.6 告警列表

告警规则名称	告警级别	告警条件	自动解除条件	告警失效条件	是否允许手动解除	告警邮件发送频率
数据目录读写错误 PathIOError	重要	数据目录状态为坏盘	数据目录状态为正常	数据目录被移除, 或数据目录所在服务器被移除	是	每天 1 次
数据目录所在磁盘写入速度变慢 DiskWriteSlow	警告	数据目录所在磁盘写入速度变慢	数据目录所在盘恢复正常	数据目录被移除, 或数据目录所在服务器被移除	是	每天 1 次
许可证即将到期 LicenseWillExpire	警告	当前时间(告警模块所在服务器的系统时间)距离最后导入的许可证的到期时间 $\leq 15$ 天且 $> 0$ 天	当前时间(告警模块所在服务器的系统时间)距离告警许可证的到期时间 $> 15$ 天	许可证过期, 或导入新的许可证(不同 id 的许可证)	是	每天 1 次
许可证过期 LicenseExpired	严重	当前时间(告警模块所在服务器的系统时间)距离最后导入的许可证的到期时间 $\leq 0$ 天	当前时间(告警模块所在服务器的系统时间)距离告警许可证的到期时间 $> 0$ 天	导入新的许可证(不同 id 的许可证)	是	发送 1 次
许可证维保即将到期 LicenseMaintenanceWillExpire	警告	当前时间(告警模块所在服务器的系统时间)距离最后导入的许可证的维保到期时间 $\leq 15$ 天且 $> 0$ 天	当前时间(告警模块所在服务器的系统时间)距离告警许可证的维保到期时间 $> 15$ 天	许可证过期, 或导入新的许可证(不同 id 的许可证)	是	每天 1 次
许可证过保 LicenseMaintenanceExpired	警告	当前时间(告警模块所在服务器的系统时间)距离最后导入的许可证的维保到期时间 $\leq 0$ 天	当前时间(告警模块所在服务器的系统时间)距离告警许可证的维保到期时间 $> 0$ 天	导入新的许可证	是	发送 1 次
试用期即将到期 TrialVersionWillExpire	警告	当前未导入生效的许可证, 并且当前时间(告警模块所在服务器的系统时间)距离试用期过期时间 $\leq 15$ 天且 $> 0$ 天	无解除条件, 只能手动解除	导入新的许可证	是	每天 1 次
资源用量接近使用上限 ResourceUsageApproachingLimit	重要	本地卷总容量 $\geq$ 许可证容量的 80%	本地卷总容量 $<$ 许可证容量的 75%	导入新的许可证	是	发送 1 次

告警中的告警条数接近上限 AlarmNumberApproachingLimit	严重	告警中的告警条数>=8000	告警条数<7500	无	是	每天1次
告警邮件发送失败 FailToSendAlarmEmail	严重	告警邮件发送失败	告警邮件发送成功	邮件配置被删除，或邮件发送设置为disable	是	每天1次
配额使用率超阈值 CapacityQuotaUsageExceedsThreshold	警告	系统、服务器或数据目录关联磁盘的Path_Cap_Quota_Rate>=80% <b>说明：</b> 数据目录未设置容量配额，则按容量配额=磁盘总容量计算。	system层级、服务器或数据目录关联磁盘的Path_Cap_Quota_Rate<75% <b>说明：</b> 数据目录未设置容量配额，则按容量配额=磁盘总容量计算。	数据目录处于移除中	是	每天1次
配额用尽 CapacityQuotaUsageApproachLimit	严重	系统对应磁盘总配额使用率>=95%	系统对应磁盘总配额使用率<90%	服务器或数据目录被移除	是	每天1次
配额用尽 CapacityQuotaUsageApproachLimit	警告	服务器对应磁盘总配额>=95%，或数据目录对应磁盘配额使用率>=95%	服务器对应磁盘总配额<90%，或数据目录对应磁盘配额使用率<90%	服务器或数据目录被移除	是	每天1次
磁盘使用率超阈值 DiskUsageExceedsThreshold	警告	系统、服务器或数据目录关联磁盘的Path_Rate>=80%	system层级、服务器或数据目录关联磁盘的Path_Rate<75%	数据目录处于移除中	是	每天1次
可用故障域数量不满足卷写入要求 InsufficientFDForLUNToWrite	警告	可用故障域数量及健康数据目录数量不满足卷的最小副本数要求	可用故障域数量及健康数据目录数量满足卷的最小副本数要求	卷禁用或者卷删除	是	每天1次
数据目录健康状态变为警告 DiskPathHealthStatusWarning	警告	数据目录健康状态变为警告	数据目录健康状态恢复正常	<ul style="list-style-type: none"> <li>● 数据目录被移除</li> <li>● 数据目录健康状态变为“Error”</li> </ul>	是	每天1次
数据目录健康状态变为错误 DiskPathHealthStatusError	重要	数据目录健康状态变为错误	数据目录健康状态恢复正常	数据目录被移除。	是	每天1次
数据服务健康状态变为警告 DataServiceHealthStatusWarning	警告	数据服务健康状态变为警告	数据服务健康状态恢复正常	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> <li>● 数据目录健康状态变为“Error”</li> </ul>	是	每天1次

数据服务健康状态变为错误 DataServiceHealthStatusError	重要	数据服务健康状态变为错误	数据服务健康状态恢复正常	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> </ul>	是	每天 1 次
协议解析服务异常 ProtocolServiceAbnormal	重要	协议解析服务异常	协议解析服务恢复正常	服务器被移除	是	每天 1 次
故障域状态变为警告 FaultDomainWarning	警告	故障域状态变为警告	故障域状态恢复正常	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> <li>● 故障域健康状态变为“Error”</li> </ul>	是	每天 1 次
故障域状态变为错误 FaultDomainError	重要	故障域状态变为 Error	故障域状态恢复正常	<ul style="list-style-type: none"> <li>● 服务器被移除</li> <li>● 数据目录被移除</li> </ul>	是	每天 1 次