Using Kaspersky Security Center OpenAPI

This article describes how to use Kaspersky Security Center OpenAPI methods to automate the deployment and use of Kaspersky Security for Virtualization 6.1 Light Agent in multitenancy mode.

For a detailed description of Kaspersky Security Center OpenAPI functions, refer to the **kscopenapi.chm** file, which is located in the Kaspersky Security Center installation folder; the default installation folder is C:\Program Files (x86)\Kaspersky Lab\Kaspersky Security Center.

Kaspersky Security Center OpenAPI usage scenarios

Moving a virtual machine to a virtual Kaspersky Security Center Administration Server

You can use this procedure to move virtual machines to a virtual Administration Server. The virtual machines must satisfy the following conditions:

- Kaspersky Security Center Network Agent must be installed on the virtual machine, and the address of the primary Administration Server must be specified in Network Agent settings.
- The virtual machine must have a unique ID.
- The virtual machine must be located on the primary Kaspersky Security Center Administration Server (for example, in the Unassigned devices group).

To move a virtual machine to the virtual Kaspersky Security Center Administration Server:

1. Authorize on the primary Kaspersky Security Center Administration Server using the Login method.
2. Get a list of virtual machines by calling the HostGroup.FindHosts method with the following fields:
   - **KLHST_WKS_DN** – display name of the virtual machine
   - **KLHST_WKS_HOSTNAME** – unique ID of the virtual machine
   - **KLHST_WKS_IP_LONG** – IP address of the virtual machine in the long format

   and a filter (KLHST_WKS_IP_LONG = **vmIPLong**), which gives you only the virtual machine with the necessary **vmIPLong** IP address.
3. In the resulting list, find your virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values of the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.
4. Create a Change Administration Server task and remember its **taskId**:
   a. In the **NewKscServerAddress** field, specify a string of the **MainKscServerAddress/VirtualServerName** form.
   b. Assign the values that you got at step 3: **HostDispName** = **KLHST_WKS_DN** and **HostName = KLHST_WKS_HOSTNAME**.
   c. Specify the display name of the task in the **DisplayName** field.
5. Start the task by specifying its **taskId** obtained at the previous step.
6. Wait for the task to complete, periodically querying its status.

Distributing the installation package to the virtual Administration Server

To distribute the installation package to the virtual Administration Server:

1. Use the PackagesApi.GetPackages method to find the package ID (**KLPKG_NPI_PKGID**) for application **KLPKG_NPI_PRODUCT_NAME** and version **KLPKG_NPI_PRODUCT_VERSION**.
2. Use the VServers.GetVServers method to find the **KLVSRV_ID** of the virtual Administration Server corresponding to the display name **KLVSRV_DN**.

3. Start the distribution of the package to the virtual Administration Server using the PackagesApi.RetranslateToVServerAsync method. The method returns an asynchronous operation ID.
4. Check the status of the asynchronous operation by its ID obtained at the previous step by calling AsyncActionStateChecker.CheckActionState .

Installing the application on a virtual machine

This procedure allows you to remotely install the application on a virtual machine if the following conditions are satisfied:

- Kaspersky Security Center Network Agent is installed on the virtual machine.
- The virtual machine is visible from the virtual Administration Server (for example, in the Unassigned devices group).
- The virtual Administration Server contains installation packages for the Network Agent and Kaspersky Security for Virtualization 6.1 Light Agent. If there packages are not present, you need to first create them or transfer them to the relevant virtual server.

To install Kaspersky Security for Virtualization 6.1 Light Agent:

1. Log in to the Administration Server on which you will run the remote installation task for the application. Use the Login method for authorization on the main Administration Server, and for authorization on the virtual Administration Server, use the Login-VirtualServer method.
2. Get a list of virtual machines by calling the HostGroup.FindHosts method with the following fields:
   - **KLHST_WKS_DN** – display name of the virtual machine
   - **KLHST_WKS_HOSTNAME** – unique ID of the virtual machine
   - **KLHST_WKS_IP_LONG** – IP address of the virtual machine.
3. In the resulting list, find your virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values of the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.
4. Use the PackagesApi.GetPackages method to find the installation package ID (**KLPKG_NPI_PKGID**) for Network Agent

   KLPKG_NPI_PRODUCT_NAME = 1103 and version **KLPKG_NPI_PRODUCT_VERSION =**
   1.0.0.0 (see the PackagesIds table).
5. Use the PackagesApi.GetPackages method to find the installation package ID (**KLPKG_NPI_PKGID**) of Kaspersky Security for Virtualization 6.1 Light Agent.
   - To install the application on a Windows virtual machine, you need a package with **KLPKG_NPI_PRODUCT_NAME** = KSVLA and **KLPKG_NPI_PRODUCT_VERSION** = 5.2.0.0 (see the PackagesIds table).
   - To install the application on a Linux virtual machine, you need a package with **KLPKG_NPI_PRODUCT_NAME** = kesl and **KLPKG_NPI_PRODUCT_VERSION** = 12.1.0.0 (see the PackagesIds table).
6. Use the HostGroup.FindGroups or HostGroup.GroupIdGroups method to specify the administration group to which you want to move the virtual machine after installing the application.
7. Create a remote installation task and remember its **taskId**:

   a. To **KlNagentPackageId**, assign the **KLPKG_NPI_PKGID** value that you got at step 4.
   b. To **ProductPackageId**, assign the **KLPKG_NPI_PKGID** value that you got at step 5.
   c. Assign the values that you got at step 3: **HostDispName** = **KLHST_WKS_DN** and **HostName = KLHST_WKS_HOSTNAME** .
   d. In the **HostOsUserLogin** and **HostOsUserPassword** fields, specify the user name and password of the user that will be used to perform the installation.
   e. In the **GroupToMoveHostId** field, specify the administration group to which you want to move the virtual machine that you got at step 6.
   f. Specify the display name of the task in the **DisplayName** field.

8. Start the task, specifying its **taskId** that you got at the previous step.
9. Wait for the task to complete, periodically querying its status.

Removing the application from the virtual machine

To remove Kaspersky Security for Virtualization 6.1 Light Agent using the Network Agent:

1. Log in to the Administration Server on which you will run the remote installation task for the application. Use the Login method for authorization on the main Administration Server, and for authorization on the virtual Administration Server, use the Login-VirtualServer method.
2. Get a list of virtual machines by calling the HostGroup.FindHosts method with the following fields:
   - **KLHST_WKS_DN** – display name of the virtual machine
   - **KLHST_WKS_HOSTNAME** – unique ID of the virtual machine
   - **KLHST_WKS_IP_LONG** – IP address of the virtual machine.
3. In the resulting list, find your virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values of the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.
4. Create a remote removal task and remember its **taskId**:
   a. Assign the values that you got at step 3: **HostDispName** = **KLHST_WKS_DN** and **HostName = KLHST_WKS_HOSTNAME**.
   b. In the **HostOsUserLogin** and **HostOsUserPassword** fields, specify the user name and password of the user that will be used to perform the removal.
   c. Specify information about the application being removed and its version in the following fields:

      ProductName (for example, KSVLA) and **ProductVersion** (for example, 5.2.0.0).
   d. Specify the display name of the task in the **DisplayName** field.
5. Start the task, specifying its **taskId** that you got at the previous step.
6. Wait for the task to complete, periodically querying its status.

Removing a virtual machine from the protected infrastructure

Before removing a virtual machine, you must remove the application installed on the virtual machine.

To remove a virtual machine from the protected infrastructure:

1. Use the Login-VirtualServer method to authorize on the Kaspersky Security Center virtual Administration Server where your virtual machine is located.
2. Get a list of virtual machines by calling the HostGroup.FindHosts method with the following fields:
   - **KLHST_WKS_DN** – display name of the virtual machine
   - **KLHST_WKS_HOSTNAME** – unique ID of the virtual machine
   - **KLHST_WKS_IP_LONG** – IP address of the virtual machine.
3. In the resulting list, find your virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values of the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.
4. If you want to remove a virtual machine from the Managed devices group and move it to the Unassigned devices group:
   a. Get the **id** of the Unassigned devices group by calling the HostGroup.GroupIdUnassigned method.
   b. Call the method for moving the virtual machine to the administration group. To do so, in the **hostId** field, set the **KLHST_WKS_HOSTNAME** value that you obtained at step 3, and in the **groupId** field, set the **id** value that you got at step 4a.
5. If you want to completely delete the virtual machine, you need to call virtual machine deletion method. To do so, as the **hostId**, pass the **KLHST_WKS_HOSTNAME** value that you got at step 3.

Used methods of Kaspersky Security Center OpenAPI

Authorization on the primary Kaspersky Security Center Administration Server

To use the Kaspersky Security Center OpenAPI, you must authorize on the Kaspersky Security Center Administration Server by calling the **login** method:

```
POST https://MainKscServerIpAddress:Port/api/v1.0/login HTTP/1.1

Authorization: KSCBasic user="Base64UserKscLogin",
pass="Base64UserKscPassword", internal="0"
Content-Type: application/json
Content-Length: 2

Body: {}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.

- **Base64UserKscLogin** is a user name on the primary Administration Server in Base64 encoding.
- **Base64UserKscPassword** is the password of the user on the primary Administration Server in Base64 encoding.

Authorization on the Kaspersky Security Center virtual Administration Server

```
POST https://MainKscServerIpAddress:Port/api/v1.0/login HTTP/1.1

Authorization: KSCBasic user="Base64UserVirtualKscLogin",
pass="Base64UserVirtualKscPassword", internal="0"
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2

Body: {}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64UserVirtualKscLogin** is a user name on the virtual Administration Server in Base64 encoding.
- **Base64UserVirtualKscPassword** is the password of the user on the virtual Administration Server in Base64 encoding.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding.

Managing virtual machines

Searching virtual machines on the Kaspersky Security Center Administration Server

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.FindHosts

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body:
{
    "wstrFilter": "(KLHST_WKS_IP_LONG = vmIPLong)",
    "vecFieldsToReturn": [
        "KLHST_WKS_DN",
        "KLHST_WKS_HOSTNAME"
        "KLHST_WKS_IP_LONG"
    ],
    "vecFieldsToOrder": [],
    "pParams": {
        "KLSRVH_SLAVE_REC_DEPTH": 128,
        "KLGRP_FIND_FROM_CUR_VS_ONLY": true
    },
    "lMaxLifeTime": 600
}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **wstrFilter** is the filter to be used for the virtual machine search. In this example, specify KLHST_WKS_IP_LONG = **vmIPLong**, where **vmIPLong** is the IP address of your virtual machine in the long format.
- **vecFieldsToReturn** is an array of virtual machine property names that must be returned. For a full list of fields, refer to the documentation of the method.
    - **KLHST_WKS_DN** – display name of the virtual machine
    - **KLHST_WKS_HOSTNAME** – unique ID of the virtual machine
    - **KLHST_WKS_IP_LONG** – IP address of the virtual machine.

The method returns the ID of the ChunkAccessor iterator object on the Administration Server in the following form:

```
{"strAccessor": "iteratorId"}
```

You can use this ID to get information about virtual machines using the iterator methods. When you are done working with the iterator, you must release the iterator by calling the ChunkAccessor.Release method.

Moving a virtual machine to an administration group

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.MoveHostsToGr
oup

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body:
{
    "pHostNames": [
        "hostId"
    ],
    "nGroup": groupId
}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.

- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **hostId** is the ID of the virtual machine to be moved. The **KLHST_WKS_HOSTNAME** value returned by HostGroup.FindHosts.
- **groupId** is the ID of the administration group to which you want to move the virtual machine. The **id** value returned by HostGroup.FindGroups.

Removing a virtual machine

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.RemoveHost

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"strHostName": "hostId"}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Kaspersky Security Center Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **hostId** is the ID of the virtual machine to be removed. The **KLHST_WKS_HOSTNAME** value returned by HostGroup.FindHosts.

Working with the iterator

Getting the item count of the iterator

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/ChunkAccessor.GetItemsC
ount

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"strAccessor": "iteratorId"}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.

- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **iteratorId** is the ID of the iterator returned in the **strAccessor** field by the HostGroup.FindHosts method.

This method returns the item count of the iterator:

```
{"PxgRetVal": itemsCount}
```

Getting iterator items

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/ChunkAccessor.GetItemsC
hunk

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body:
{
  "strAccessor": "iteratorId",
  "nStart": from,
  "nCount": itemsCount

}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **iteratorId** is the ID of the iterator returned in the **strAccessor** field by the HostGroup.FindHosts method.
- **from** is the starting index of the collection item from which items must be returned. The first item has index 0.
- **itemsCount** is the number of items to return. If you want to return all items, you can use the **itemsCount** value returned by the ChunkAccessor.GetItemsCount method.

This method returns iterator items in the following form (example for calling the HostGroup.FindHosts method):

```
{ "pChunk":{
     "KLCSP_ITERATOR_ARRAY":[
     {
         "type":"params",
         "value":{
             "KLHST_WKS_DN":"Host1",
             "KLHST_WKS_HOSTNAME":"7ad995e2-eb62-40e5-
9c7e5abae19979a0",
             "KLHST_WKS_IP_LONG":{
                 "type":"long",
                 "value":2130706433
             }
         }
     },
     {
         "type":"params",
         "value":{
             "KLHST_WKS_DN":"Host2",
             "KLHST_WKS_HOSTNAME":"5fb6a90c-d054-4f9b-
a3422a62949ad899",
             "KLHST_WKS_IP_LONG":{
                "type":"long",
                "value":172052763
             }
         }
     }],
     "PxgRetVal":2
 }
```

In this example, two items are returned (`"PxgRetVal":2`) with the fields specified in the **vecFieldsToReturn** parameter of the HostGroup.FindHosts request.

Releasing the iterator

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/ChunkAccessor.Release

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"strAccessor": "iteratorId"}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **iteratorId** is the ID of the iterator returned in the **strAccessor** field by the HostGroup.FindHosts method.

Managing administration groups

Searching administration groups

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.FindGroups

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body:
{
    "wstrFilter": "(name = groupName)",
    "vecFieldsToReturn": [ "id", "name" ],
    "vecFieldsToOrder": [],
    "pParams": {},
    "lMaxLifeTime": 600
}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **wstrFilter** is the filter used to seach for administration groups. In this example, specify (name = **groupName**), where **groupName** is the name of the administration group that you need.
- **vecFieldsToReturn** is an array of administration group property names that must be returned. For a full list of fields, refer to the documentation of the method.
  - **id** is the group ID.
  - **name** is the group name.

The method returns the ID of the ChunkAccessor iterator object on the Administration Server in the following form:

```
{"strAccessor": "iteratorId"}
```

You can use this ID to get information about administration groups using the iterator methods. When you are done working with the iterator, you must release the iterator by calling the ChunkAccessor.Release method.

Getting the ID of the Managed devices group

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.GroupIdGroups

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2

Body: {}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.

This method returns the group ID in the following form:

```
{"PxgRetVal": groupId}
```

Getting the ID of the Unassigned devices group

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.GroupIdUnassigned

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2

Body: {}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.

This method returns the group ID in the following form:

```
{"PxgRetVal": groupId}
```

Getting a list of virtual Administration Servers

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/VServers.GetVServers

Content-Type: application/json
Content-Length: 2

Body: {}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.

If successful, the method returns an array of virtual Administration Servers in the following form:

```
{"PxgRetVal": [
    {
        "type": "params",
        "value": {
            "KLVSRV_CREATED": {
                "type": "datetime",
                "value": "2020-06-15T09:41:18Z"
            },
            "KLVSRV_DN": "SomeVirtualServer",
            "KLVSRV_ENABLED": true,
            "KLVSRV_GROUPS": 8,
            "KLVSRV_GRP": 0,
            "KLVSRV_HST_UID": "VSRV08172e1f-4057-4579-
   89c4d5e6256d8ad2",
            "KLVSRV_ID": 1,
            "KLVSRV_LIC_ENABLED": true,
            "KLVSRV_SUPER": 7,
            "KLVSRV_UID": "VSRV08172e1f-4057-4579-89c4-d5e6256d8ad2",
            "KLVSRV_UNASSIGNED": 11
        }
    }]
}
```

Important parameters:

- **KLVSRV_DN** is the name of the virtual Administration Server.
- **KLVSRV_ID** is the ID of the virtual Administration Server.

Managing installation packages

Getting a list of installation packages

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/PackagesApi.GetPackages

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2

Body: {}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.

If successful, the method returns an array of packages in the following form:

```
{"PxgRetVal": [
    {
        "type": "params",
        "value": {
            "KLPKG_NPI_CREATION_TIME": {
                "type": "datetime",
                "value": "2019-10-08T17:30:40Z"
            },
            "KLPKG_NPI_MODIF_TIME": {
                "type": "datetime",
                "value": "2019-10-08T17:30:40Z"
            },
            "KLPKG_NPI_NAME": "Kaspersky Security for Virtualization
   5.2 Light Agent.",
            "KLPKG_NPI_PACKAGE_PATH":
   "\\\\DESKTOPMI1CJOJ\\KLSHARE\\Packages\\KSVLA_5.2.27.1202",
            "KLPKG_NPI_PKGID": 1,
            "KLPKG_NPI_PRODUCT_DISPL_NAME": "Kaspersky Security for
   Virtualization 5.2 Light Agent",
            "KLPKG_NPI_PRODUCT_DISPL_VERSION": "5.2.27.1202",
            "KLPKG_NPI_PRODUCT_NAME": "KSVLA",
            "KLPKG_NPI_PRODUCT_VERSION": "5.2.0.0",
            "KLPKG_NPI_SIZE": {
                 "type": "long",
                 "value": 217427607
            },
            "KLPKG_NPI_SS_DESCR":
   "KSVLA_5.2.27.1202\\exec\\ss_install.xml|1"
        }
    }, ....
    ]
}
```

Important parameters:

- **KLPKG_NPI_PKGID** is the ID of the installation package.
- **KLPKG_NPI_PRODUCT_NAME** is the name of the application that installation package installs.
- **KLPKG_NPI_PRODUCT_VERSION** is the version of the application that the installation package installs.

Installation packages of Kaspersky Security Center Network Agent and the Kaspersky Security application

To identify the packages of Network Agent and Kaspersky Security for Virtualization 6.1 Light Agent, when calling Kaspersky Security Center OpenAPI methods, you can use the following table:

Package property name	Network Agent	Windows LA 5.2	KES for Linux 12.0
KLPKG_NPI_PRODUCT_NAME	1103	KSVLA	kesl
KLPKG_NPI_PRODUCT_VERSION	1.0.0.0	5.2.0.0	12.1.0.0

Distributing the installation package to the virtual Administration Server

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/PackagesApi.Retranslate
ToVServerAsync
```

```
Content-Type: application/json
Content-Length: BodyLength

Body:
{
    "nPackageId": packageId,
    "nVServerId": VirtualServerId,
    "pOptions":
    {
        "KLPKG_CREATE_STANDALONE_PRODS": true,
        "KLPKG_CREATE_STANDALONE_NAGT": true,
        "KLPKG_USE_LANGUAGE_TAG": true,
        "KLPKG_TYPE": 1,
        "KLPKG_LAZY_RETRANSLATION": false
    }
}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **BodyLength** is the length of the JSON request body in bytes.
- **packageId** is the ID of the installation package to be distributed to the virtual Server. **KLPKG_NPI_PKGID** value returned by the PackagesApi.GetPackages method.
- **VirtualServerId** is the ID of the virtual Administration Server to which you want to distribute the package. **KLVSRV_ID** value returned by the VServers.GetVServers method.

If successful, the method returns the asynchronous operation ID in the following form:

```
{"PxgRetVal": "asyncActionId"}
```

To check if the asynchronous package transfer operation is completed, call the AsyncActionStateChecker.CheckActionState method.

Checking the status of an asynchronous operation

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/AsyncActionStateChecker
.CheckActionState

Content-Type: application/json
Content-Length: BodyLength

Body: {"wstrActionGuid": "asyncActionId"}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **BodyLength** is the length of the JSON request body in bytes.

- **asyncActionId** is the asynchronous operation ID returned by the PackagesApi.RetranslateToVServerAsync method.

If successful, the method returns the state of the asynchronous operation in the following form:

```
{
    "bFinalized": true,
    "bSuccededFinalized": true,
    "lStateCode": 0,
    "pStateData": {},
    "lNextCheckDelay": 0
}
```

where:

- **bFinalized** is the state of the operation. `true` means the operation is complete.
- **bSuccededFinalized** indicates if the operation was completed successfully. `true` means the operation was completed successfully.

Managing tasks

Creating a Change Administration Server task

This task lets you change the Administration Server that is managing a virtual machine.

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.AddTask

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"pData": taskParams}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Kaspersky Security Center Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **taskParams** are parameters of the Change Administration Server task:

```
{"pData" : {
    "TASKID_PRODUCT_NAME": "1103",
    "TASK_ADDITIONAL_PARAMS": {
        "type": "params",
        "value": {
            "ServerSslPorts": [ 13000 ],
            "Serverports": [ 14000 ],
            "ServerAddress": "NewKscServerAddress"
        }
    },
    "TASK_INFO_PARAMS": {
       "type": "params",
       "value": {
           "DisplayName": "Reconnect to vKSC",
           "HostList": [{
               "type": "params",
               "value": {
                   "HostDispName": "Host1",
                   "HostName": "7ad995e2-eb62-40e5-9c7e-
5abae19979a0"
                }
            }]
        }
    },
    "TASKSCH_RUN_MISSED_FLAG": true,
    "TASKID_VERSION": "1.0.0.0",
    "TASKSCH_TYPE": 0,
    "TASK_NAME": "KLNAG_TASK_RECONNECTION",
    "TASKID_COMPONENT_NAME": "86"
} }
```

where:

- **ServerSslPorts** is an array of SSL ports of the new Administration Server. By default, one port number **13000** is specified.
- **Serverports** is an array of ports of the new Administration Server. By default, one port number **14000** is specified.
- **ServerAddress** is the IPv4 address or the fully qualified domain name (FQDN) of the new Administration Server. If the virtual machine will be managed by a virtual Administration Server, the value must be specified as **MainKscServerAddress/VirtualServerName**.
- **DisplayName** is the display name of the task.
- **HostDispName** is the name of the virtual machine that you want to move to a different Administration Server. **KLHST_WKS_DN** value returned by the HostGroup.FindHosts method.
- **HostDispName** is the unique ID of the virtual machine that you want to move to a different Administration Server. Value **KLHST_WKS_HOSTNAME** value returned by the HostGroup.FindHosts method.

If successful, the method returns the ID of the created task in the following form:

```
{"PxgRetVal": "taskId"}
```

Creating a remote application installation task

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.AddTask
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"pData": taskParams}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.

- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server.
- **BodyLength** is the length of the JSON request body in bytes.
- **taskParams** are the settings of the remote installation task:

```
{"pData": {
    "TASKID_PRODUCT_NAME": "1093",
    "TASK_ADDITIONAL_PARAMS": {
        "type": "params",
        "value": {
            "KLTSK_RI_USE_SHARE": true,
            "KLTSK_RI_USE_SHARE_SRV": true,
            "KLTSK_RI_PACKAGES_IDS": [
                KlNagentPackageId,
                ProductPackageId
            ],
            "KLTSK_RI_USE_NAGENT": true,
            "KLTSK_RI_GROUP_TO_MOVE_HOST": GroupToMoveHostId,
            "klprts-TaskAccounts": [
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountAuthType": 1
                }
            },
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountUser": "HostOsUserLogin"
                }
            },
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountPassword":
"HostOsUserPassword"
                }
            }]
        }
   },
   "TASK_INFO_PARAMS": {
       "type": "params",
       "value": {
           "DisplayName": "Install KSVLA 5.2.0.0 on host",
           "HostList": [
               "type": "params",
               "value": {
                   "HostDispName": "Host1",
                   "HostName": "7ad995e2-eb62-40e5-9c7e-
5abae19979a0"
               }
           ]
       }
   },
   "TASKSCH_RUN_MISSED_FLAG": true,
   "TASKID_VERSION": "1.0.0.0",
   "TASKSCH_TYPE": 0,
        "TASK_NAME": "Remote Installation",
        "TASKID_COMPONENT_NAME": "87"
    }}
```

where:

- **KlNagentPackageId** is ID of the Network Agent installation package found by calling the PackagesApi.GetPackages method. The package must be located on the Administration Server where the virtual machine is located. To transfer the packages to the relevant Administration Server, use the PackagesApi.RetranslateToVServerAsync method.
- **ProductPackageId** is the ID of the Kaspersky Security for Virtualization 6.1 Light Agent installation package found by calling the PackagesApi.GetPackages method. The package must be located on the Administration Server where the virtual machine is located. To transfer the packages, use the PackagesApi.RetranslateToVServerAsync method.
- **GroupToMoveHostId** is the ID of the administration group to which you want to move the virtual machine after installation of the application. To find thje group, use the HostGroup.FindGroups or HostGroups.GroupIdGroups method.
- **HostOsUserLogin** is the name of the user on whose behalf the application will be installed.
- **HostOsUserPassword** is the password of the user on whose behalf the application will be installed.
- **DisplayName** is the display name of the task.
- **HostDispName** is the name of the virtual machine on which you are installing the application. **KLHST_WKS_DN** value returned by the HostGroup.FindHosts method.
- **HostName** is the unique ID of the virtual machine on which you are installing the application. **KLHST_WKS_HOSTNAME** value returned by the HostGroup.FindHosts method.

If successful, the method returns the ID of the created task in the following form:

```
{"PxgRetVal": "taskId"}
```

Creating a remote application removal task

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.AddTask

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"pData": taskParams}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if the method is called for a virtual server.
- **BodyLength** is the length of the JSON request body in bytes.
- **taskParams** are the settings of the remote removal task:

```
{"pData": {
    "TASKID_PRODUCT_NAME": "1093",
    "TASK_ADDITIONAL_PARAMS": {
        "type": "params",
        "value": {
            "klprts-TaskAccounts": [
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountAuthType": 1
                }
            },
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountUser": "HostOsUserLogin"
                }
            },
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountPassword":
"HostOsUserPassword"
                }
            }],
            "KLTSK_RI_USE_SHARE_SRV": true,
            "KLTSK_RI_USE_SHARE": true,
            "ProductVersion": "5.2.0.0",
            "KLTSK_RI_USE_NAGENT": true,
            "ProductName": "KSVLA",
            "UninstallType": 0
        }
    },
    "TASK_INFO_PARAMS": {
        "type": "params",
        "value": {
            "DisplayName": "Deinstall KSVLA 5.2.0.0 on host",
            "HostList": [
                "HostDispName": "Host1",
                "HostName": "7ad995e2-eb62-40e5-9c7e-
5abae19979a0"
            ]
        }
    },
    "TASKSCH_RUN_MISSED_FLAG": true,
    "TASKID_VERSION": "1.0.0.0",
    "TASKSCH_TYPE": 0,
    "TASK_NAME": "Remote Deinstallation",
    "TASKID_COMPONENT_NAME": "87"
}}
```

where:

- **HostOsUserLogin** is the name of the user on whose behalf the application will be removed.
- **HostOsUserPassword** is the password of the user on whose behalf the application will be removed.
- **ProductVersion** is the version of the application to be removed.
- **ProductName** is the name of the application to be removed.
- **DisplayName** is the display name of the task.
- **HostDispName** is the name of the virtual machine on which you are removing the application. **KLHST_WKS_DN** value returned by the HostGroup.FindHosts method.

- **HostName** is the unique ID of the virtual machine on which you are removing the application. **KLHST_WKS_HOSTNAME** value returned by the HostGroup.FindHosts method.

If successful, the method returns the ID of the created task in the following form:

```
{"PxgRetVal": "taskId"}
```

Running a task

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.RunTask

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"strTask": "taskId"}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server. **BodyLength** is the length of the JSON request body in bytes.
- **taskId** is the task ID received after its creation.

Viewing task status

```
POST https://MainKscServerIpAddress:Port/api/v1.0/
Tasks.GetTaskStatistics

X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength

Body: {"strTask": "taskId"}
```

where:

- **MainKscServerIpAddress** is the IPv4 address or fully qualified domain name (FQDN) of the primary Kaspersky Security Center Administration Server.
- **Port** is the Kaspersky Security Center OpenAPI port on the primary Administration Server. The default port number is **13299**.
- **Base64VirtualKscName** is the name of the virtual Administration Server in Base64 encoding. The `X-KSC-VServer` header is required if you are calling the method for a virtual Administration Server. **BodyLength** is the length of the JSON request body in bytes.
- **taskId** is the task ID received after its creation.

If successful, the method returns the task status in the following form:

```
{"PxgRetVal": {
   "1": "notDistributed",
   "2": "running",
   "4": "succeeded",
   "8": "warning",
   "16": "failed",
   "32": "scheduled",
   "64": "paused",
   "GNRL_COMPLETED_PERCENT": 0,
   "KLTSK_NEED_RBT_CNT": 0
}}
```

where:

- **notDistributed** is the number of virtual machines on which the task has not been started yet
- **running** is the number of virtual machines on which the task is running
- **succeeded** is the number of virtual machines on which the task has successfully finished
- **warning** is the number of virtual machines on which the task finished with a warning
- **failed** is the number of virtual machines on which the task finished with an error
- **scheduled** is the number of virtual machines on which the task is scheduled to run
- **paused** is the number of virtual machines on which the task is paused