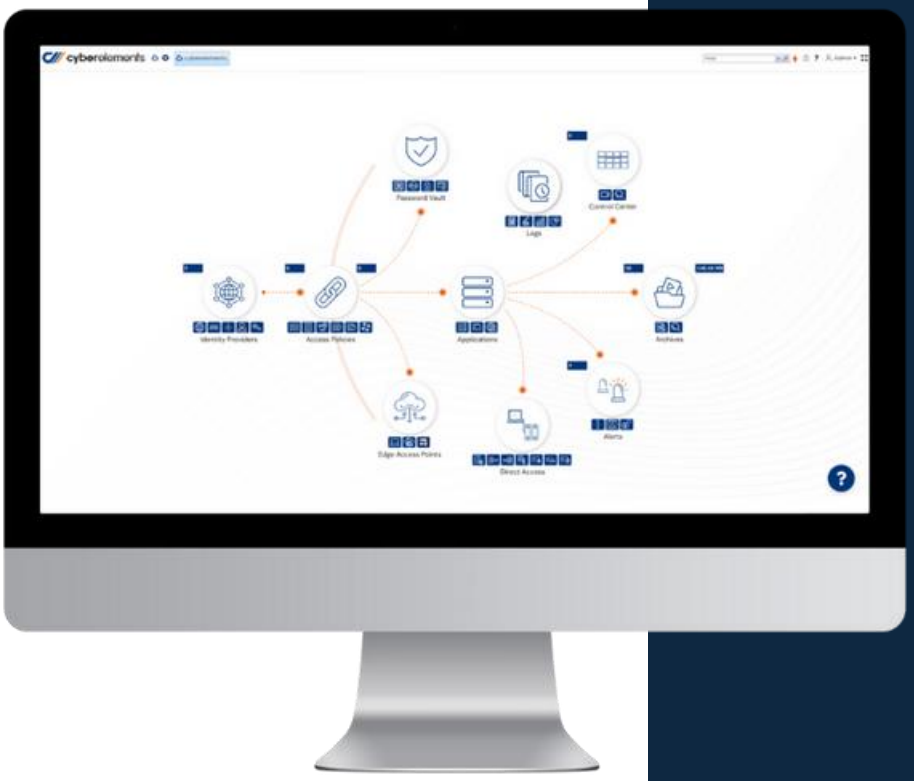
cyberelements Cleanroom

cyberelements Cleanroom API

- version 4.6.1



www.systancia.com

Ref.:	EN_cyberelements-Cleanroom_MA-0003_cyberelements Cleanroom API_rev.1.05
Version:	1.05
Product:	**cyber**elements Cleanroom
Date:	2026-03-31

Summary:

This manual covers the operation of the **cyber**elements Cleanroom API version 4.6.1.

TABLE OF CONTENTS

1 General information

The **cyber**elements Cleanroom 4.6.1 API is a REST API. It is accessible via the IP address (HTTPS) of the Mediation Controller server on which the **cyber**elements Cleanroom site is configured.

The data sent and received by the API is in JSON format.

1.1 Implementation of the API

The API must be enabled by modifying the api/enabled parameter in the /etc/ipdiva/care/djangosettings.ini file located on the Mediation Controller server.

```
[api]
enabled = True
```

After that, you must restart the *apache2* service on the Mediation Controller server for the configuration change to take effect.

1.2 Request types

The standard request methods (GET, POST, DELETE, PUT, PATCH) can be used with this API.

However, it is necessary to know that:

- A modification with PUT always replaces all the data of the object, except those which are not accessible by the API.
- URLs always end with the slash character.

Here is the list of possible main actions:

Action	Request example
Retrieve a list	GET publicapi/<ORGANIZATION>/<TYPEOBJECT>/
Retrieve an object	GET publicapi/<ORGANIZATION>/<TYPEOBJECT>/<ID>/
Create an object	POST publicapi/<ORGANIZATION>/<TYPEOBJECT>/
Delete an object	DELETE publicapi/<ORGANIZATION>/<TYPEOBJECT>/<ID>/
Modify an object	PUT publicapi/<ORGANIZATION>/<TYPEOBJECT>/<ID>/
Modify an object partially	PATCH publicapi/<ORGANISATION>/<TYPEOBJET>/<ID>/

Each section of this document describing the elements that can be manipulated via this API will list the various URLs that can be used and specify the available methods. Unless

otherwise specified, the five request methods mentioned above can be used for the elements in question.

1.3 Error management

If the call was successful, an HTTP success code is returned. It can vary depending on the type of request. For example, if an object is retrieved, the code 200 is received. In case of deletion, code 204 (No Content) is returned.

In case of error, an HTTP error code is returned. Among the possible response codes, the most common are:

- A 400 "Bad Request" code is returned if the request does not meet certain requirements, for example, when a required parameter has not been specified or when a field that cannot be empty has been left blank
- Code 403 "Forbidden" is returned if authentication information is missing or invalid
- Code 405 "Method Not Allowed" is returned when the method used for a request is not supported for that URL. Most requests supported by the REST API accept all methods described in the Request types section, but this is not the case for all of them. The sections of this document describing how to manipulate the various **cyber**elements Cleanroom elements mention exceptions to this rule

The format of the response to a request varies depending on the type of error. In general, JSON data is sent with an indication of the cause of the error. However, in some cases, a 500 error may be returned. This is the case, for example, if the requested operation violates a database integrity constraint (duplicate values that are supposed to be unique, etc.). In such cases, details of the error that occurred are available in the Apache2 service log files on the contacted Mediation Controller server ("/var/log/apache2/*_error.log" files).

1.4 Overview of concepts and examples

In the following sections of this document, the **cyber**elements Cleanroom elements that can be manipulated via the REST API will be presented one by one, each in a dedicated section. These sections will contain a number of elements, including:

- The list of API URLs associated with the element in question.
- A list of request methods (GET, POST, etc.) that can be used for these URLs, if they differ from the standard (see Request types)
- A description of the *Data format* for the element in question, in the form of an insert containing an example JSON object representing a typical element, followed by a table explaining each of the object's properties.
- One or more examples of requests sent to the REST API, in the form of an insert containing an example written in *Python* code using the *requests* library.

2 Administration console API

2.1 Authentication

Every API call must be authenticated. Authentication is based on the "Authorization" HTTP header, which must be included in all requests to the **cyber**elements Cleanroom API. This header must contain a token previously obtained by a call to a function dedicated to the publicapi/api-auth URL.

For the administration console API, the credentials to be provided are those of an administrator for the organization's default local domain.

The configuration of the organization's default local domain applies: too many failed authentication attempts may result in the account being locked out, depending on the configuration.

Additionally, only a basic login/password configuration can be used: if certificate-based authentication is required on this domain, access to the REST API is not possible. The same applies to other features that modify authentication, such as OTPs.

Example of obtaining the token:

```
import requests

r = requests.post('https://<mediation_server>/publicapi/api-auth', json={
    'login': 'admin',
    'password': 'secret',
    'org': 'organisation'
})
data = r.json()
try:
    id = data['id']
    print("Authentication succeeded")
except KeyError:
    print("Authentication failed")
    sys.exit(1)
```

The request must include the following parameters, using JSON syntax:

Parameters	Description
login	The name of the organization's local domain administrator.
password	The password of the administrator of the organization's local domain.
org	The target organization for requests.

The response must contain a JSON object with an "id" field that contains the expected token. This token must then be included in the "Authorization" header for subsequent requests. The examples in the following sections will include this header.

If the response does not contain an "id" field, authentication has failed.

2.2 Categories

URL:

/publicapi/<ORGANIZATION>/categories/

/publicapi/<ORGANIZATION>/categories/<ID>/

/publicapi/<ORGANIZATION>/categoriesbyname/< NAME>/

Data format:

```
{
"name": "name",
"cat_id": 1
}
```

Parameter	Description
name	Category Name
cat_id	Category ID (read-only)

Example of creation:

```
data = {'name':'testapi'}

requests.post('https://<mediation_server>/publicapi/<org>/categories/', json=data,
headers={'Authorization': id})
```

Example of deletion:

```
requests.delete('https://<mediation_server>/publicapi/<org>/categories/1246/',head
ers={'Authorization':id})
```

2.3 Sites

URL:

/publicapi/<ORGANIZATION>/sites/

/publicapi/<ORGANIZATION>/sites/<ID>/

/publicapi/<ORGANIZATION>/sitesbyname/<NAME>/

Data format:

```
{
    "site_id": 1,
    "name": "site3",
    "description": "default site",
    "gateways": [
        {"gateway_name": "gw-name-1", "gateway_usage": "master"}
    ],
    "gateways_html5": ["gw-h5-name-1"]
}
```

Parameter	Description
site_id	Site ID (read-only)
name	Site Name
description	Site description
gateways	The list of associations between the site and all Edge Gateways. Each Edge Gateway in the organization is listed here, along with its status within the site: • "master" if the Edge Gateway is active • "slave" if the Edge Gateway is passive • "nothing" if the Edge Gateway is unused
gateways_html5	The list of HTML5 Edge Gateway names associated with the site

Example of retrieving the list of sites:

```
requests.get('https://<mediation_server>/publicapi/<org>/sites/',
headers={'Authorization':id})
```

Answer:

```
[
    {
        "site_id":1,
        "name":"site3",
        "description":"default site",
        "gateways":[
            {"gateway_name": "gw-name-1","gateway_usage": "master"},
            {"gateway_name": "gw-name-2","gateway_usage": "nothing"}
        ],
        "gateways_html5": ["gw-h5-name-1", "gw-h5-name-2"]
    }
]
```

Example of modification of a site:

```
data = {
    "name": "site_3",
    "description": "default site",
    "gateways": [
        {"gateway_name": "gw-name-3", "gateway_usage": "slave"}
    ]
    "gateway_html5": ["gw-h5-name-1"],
}

requests.patch('https://<mediation_server>/publicapi/<org>/sites/1/', json=data,
headers={"Authorization":id})
```

Note: When creating or modifying sites, only the Edge Gateways explicitly specified in the "gateways" parameter are affected by the request. The statuses of unspecified Edge Gateways remain unchanged.

2.4 Edge Gateways

URL:

/publicapi/<ORGANIZATION>/gateways/

/publicapi/<ORGANIZATION>/gateways/<ID>/

/publicapi/<ORGANIZATION>/gatewaysbyname/<NAME>/

Note: It is not possible to modify Edge Gateways via the API, whether using PUT or PATCH requests.

Data format:


```
{
    "gateway_id": 1,
    "name": "test gateway",
    "description": "",
    "fqdn": "TEST_FQDN",
    "archive_path": "/var/lib/ipdiva/carerecord/archives",
    "ssh_archive_path": "/var/ipdiva/care/sshrecord",
    "status": {
        "online": true,
        "version": "8.9.1.1096"
    },
    "token": ""
}
```


Parameter	Description
gateway_id	Edge Gateway ID (read-only)
name	Name of the Edge Gateway. Must be unique
description	Description of the Edge Gateway *Note: This field is currently required and cannot be left blank*
fqdn	FQDN (*Fully Qualified Domain Name*) of the Edge Gateway
archive_path	Path to the location of the graphics session archives on the Edge Gateway (read-only)
ssh_archive_path	Path to the SSH session archive location on the Edge Gateway (read-only)
status	Edge Gateway status, indicating whether it is online and its version number (read-only)
token	Unused parameter (read-only)

Example of creation:

```
data = {
    'name': 'test gateway',
    'fqdn': 'TEST_FQDN',
    'description': 'test description'
}

requests.post('https://<mediation_server>/publicapi/<org>/gateways/', json=data,
headers={"Authorization":id})
```

Note: It is currently not possible to use the feature for creating pairing tokens via the REST API.

2.5 HTML5 Edge Gateways

URL:

/publicapi/<ORGANIZATION>/html5gateways/

/publicapi/<ORGANIZATION>/html5gateways/<ID>/

/publicapi/<ORGANIZATION>/html5gatewaysbyname/<NAME>/

Data format:

```
{
    "gateway_html5_id": 1,
    "name": "t3182-html5",
    "description": "",
    "url": "/HTML5_HTTP",
    "protocol": "http",
    "status": {
        "online": true,
        "version": "8.9.1.1096"
    },
    "token": ""
}
```

Parameter	Description
gateway_html5_id	The HTML5 Edge Gateway ID (read-only)
name	Name of the HTML5 Edge Gateway. Must be unique
description	Description of the HTML5 Edge Gateway *Note: This field is currently required and cannot be left blank*
url	URL of the HTML5 Edge Gateway
protocol	Protocol to use for communication with the HTML5 Edge Gateway. Can be either "http" or "websocket"
status	Edge Gateway status, indicating whether it is online and its version number (read-only)
token	Unused parameter (read-only)

Example of creation:

```
data = {
    "name": "test gw html5",
    "description": "test description",
    "url": "/HTML5",
    "protocol": "websocket"
}

requests.post('https://<mediation_server>/publicapi/<org>/html5gateways/',
json=data, headers={"Authorization":id})
```

2.6 Applications

Applications are classified by application type and service type.

There are two types of applications: “Standard Application” and “Privileged Application”. Each of these two types can then be further categorized into several service types.

These two classifications determine the entry point for managing applications, as well as the acceptable parameters and those required by the API for their manipulation. However, all applications are based on a common set of parameters, which are grouped into a single “resource” parameter.

2.6.1 Common parameters

Since all applications are based on a common set of parameters, it is possible to list applications regardless of their service type and retrieve all these common parameters.

URL:

/publicapi/<ORGANIZATION>/resources/

/publicapi/<ORGANIZATION>/resources/<ID>/

/publicapi/<ORGANIZATION>/resourcesbyname/<NAME>/

Data format:


```
{
   "resource_id": 2,
   "name": "test rds",
   "description": "",
   "category": "https://<mediation_server>/publicapi/<org>/categories/<id>/",
   "application_type": "priv",
   "service_type": "rdp",
   "restrictable": true,
   "assistance": false,
   "token_auth": 0,
   "ask_for_comment": false,
   "video_deletable": true,
   "video_storage_duration_days": null,
   "store_hash": false,
   "register_video": true,
   "enable_SSO": "sso-fixe",
   "aliases": ["alias_name_1"],
   "motd": "",
   "server": "1.2.3.4",
   "socket": 3389,
   "dynamic": false,
   "type": "",
   "ip_list": "",
   "ip_begin": null,
   "ip_end": null,
   "ip_mask": null,
   "mask": "",
   "setwindowtitle": false
}
```

Parameter	Description
resource_id	Application ID (read-only)
name	Application name. Must be unique
description	Description of the application
category	A reference to the category to which the application belongs, in the form of a REST API URL
application_type	Application type. Possible values: "priv" and "std". Although this field is not read-only, it is not recommended to modify it in an existing application, as this may render the application unusable.
service_type	Application service type (read-only)
restrictable	The app's "may be restricted" status. Irrelevant in most use cases
assistance	Deprecated parameter
token_auth	Deprecated parameter
ask_for_comment	Enable the prompt to request a comment from the user. If enabled, the app will prompt the user for a comment upon launch.
video_deletable	Enable administrators to manually delete session logs for this application via the administration console
video_storage_duration_days	Retention period for this application's archives, in days. An empty value disables automatic deletion of archives
store_hash	Enable video integrity checks during playback
register_video	Enable video recording of app sessions. Disabling this option sets the app to "events only" recording mode and disables the generation of a video for its archives. Applies only to privileged graphical apps
enable_SSO	The application's SSO mode. Possible values: "sso", "no-sso", "askPassword" and "sso-fixe" • "sso" sets the application's SSO to "Enabled" mode • "no-sso" sets the application's SSO to "Disabled" mode • "askPassword" sets the application's SSO to "Ask" mode • "sso-fixed" sets the application's SSO to "Fixed" mode and allows aliases from the vault to be associated with this application (see "aliases" parameter)
aliases	List of alias names associated with this application. Relevant and required only if the SSO mode is "sso-fixed".

Parameter	Description
	If the vault is in *Keeper* mode, then the alias names are Keeper UIDs. *Note: It is currently not possible to associate a new Alias Keeper with an application via the **cyber**elements Cleanroom Public API*
motd	Warning message displayed to the user when they launch their app
dynamic	Enabling "dynamic application" mode. A dynamic application does not target a single address, but rather a collection of addresses, requiring the user to enter a specific address when launching the application
server	Server targeted by the application. Can be an IP address or a hostname. Relevant and required only for non-dynamic applications
socket	Port of the server targeted by the application. Relevant only for non-dynamic applications
type	Restriction type for the "dynamic" aspect of the application, determining which addresses are eligible as targets when the application is launched. Can take the following values: • "ip_list": the application can target addresses explicitly listed in the "ip_list" parameter • "ip_range": the application can target IP addresses within an address range, defined via the "ip_begin" and "ip_end" parameters • "ip_mask": the application can target IP addresses belonging to a CIDR subnet, defined via the "ip_mask" and "mask" parameters Relevant only for dynamic applications
ip_list	A comprehensive list of IP addresses that can be targeted by the application. Relevant and required only for dynamic applications with an "ip_list" restriction type
ip_begin	The first IP address in the range of IP addresses allowed as targets for the application. Relevant and required only for dynamic applications with an "ip_range" restriction type
ip_end	The last IP address in the range of IP addresses authorized as targets for the application. Relevant and required only for dynamic applications with an "ip_range" restriction type
ip_mask	The subnet's IP address, used in conjunction with the "mask" parameter to determine which addresses the application can target. Relevant and required only for dynamic applications with an "ip_mask" restriction type

Parameter	Description
mask	CIDR subnet mask, used in conjunction with the "ip_mask" parameter to determine the addresses that the application can target. Relevant and required only for dynamic applications with an "ip_mask" restriction type
setwindowtitle	Enables the application window title to be overwritten if the application is launched in windowed mode. If this option is enabled, the session window title is replaced with the address targeted by the application (the "server" parameter). Applies only to VNC and RDP applications using a *Windows* client

2.6.2 Privileged RDS applications

URL:

/publicapi/<ORGANIZATION>/rdpservices/

/publicapi/<ORGANIZATION>/rdpservices/<ID>/

/publicapi/<ORGANIZATION>/rdpservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the RDS-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:


```
{
    "service_id": 1,
    "resource": {},
    "colors_rdp": 24,
    "resolution": "fullscreen",
    "use_all_screen": false,
    "enable_CSSP": true,
    "redirect_clipboard": true,
    "plug_and_play": "",
    "redirect_audio_input": false,
    "mount_local_disk": false,
    "mount_local_printer": false,
    "enable_AUP": false,
    "connect_local_com_port": false,
    "console_mode": false,
    "remote_application": "",
    "launch_directory": "",
    "auth_domain": "",
    "ask_id": false,
    "noagent": false,
    "mstsc_arguments": "",
    "restricted_admin": false,
    "gw_on_workstation": false,
    "disable_kerberos": false,
    "kerberos_service_accounts": "",
    "custom_mstsc_options": "",
    "disable_mstsc_certificate_verification": false,
    "use_custom_app": false,
    "custom_app": "",
    "custom_app_arguments": "",
    "recorder_timeout_sec": 30,
    "broker_collection": "",
    "enable_recording_marker": true,
    "use_rds_broker": false
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
colors_rdp	The size of the color palette to be used for the application, in bits. Can be 16, 24, or 32
resolution	Resolution to use for the application window. Can be set to "640x480", "800x600", "1024x768", "1280x1024", "1600x1200", "fullscreen", or "dynamic"
use_all_screen	Use of monitors for the remote session. If this option is enabled, the remote session can use all available monitors. If it is disabled, only one monitor will be used.
enable_CSSP	Using the *Credential Security Support Provider* for the application
redirect_clipboard	Redirecting the client's clipboard to the RDP session
plug_and_play	List of *Plug and Play* devices that can be used from the workstation's RDP client to the RDP session
redirect_audio_input	Redirecting audio input from the client machine to the RDP session
mount_local_disk	Mounting the client computer's drives on the remote computer
mount_local_printer	Mounting the client computer's printers on the remote computer
enable_AUP	Enabling support for the *Applidis Universal Printing* plugin
connect_local_com_port	Enabling the COM port connection between the local and remote computers
console_mode	Enabling *Console* mode for the application
remote_application	Name of an executable file on the remote computer to be run automatically when a session is opened
launch_directory	Working directory where the executable specified in "remote_application" is located
auth_domain	Authentication domain to use for authentication on the remote workstation. If this value is empty, the authentication domain used will be that of the alias or the **cyber**elements Cleanroom domain

Parameter	Description
	associated with the user account, depending on the SSO mode used for the application
ask_id	Deprecated parameter
noagent	Enabling the *without an agent* mode for this application
mstsc_arguments	Arguments to pass to the *mstsc.exe* executable when opening a remote session. Applies only to *Windows* client computers using *mstsc*
restricted_admin	Enabling *restrictedadmin* mode for the application. This mode prevents accounts without administrator privileges on the remote workstation from accessing it.
gw_on_workstation	Enabling the *built-in Edge Gateway* feature. Cannot be enabled via the API
disable_kerberos	Disable *Kerberos* for the application. Applies only if the "noagent" option is enabled
kerberos_service_accounts	List of service accounts to use for retrieving the *Kerberos TGT*. Must be specified as a single string, with each account name separated by a comma. Applicable only if the "noagent" option is enabled and the "disable_kerberos" option is disabled.
custom_mstsc_options	Custom settings for the RDP file used to launch the application
disable_mstsc_certificate_verification	Disabling certificate verification on the client machine. Applies only to *Windows* client computers using *mstsc*
use_custom_app	Enable the use of a client program other than *mstsc*. Use in conjunction with the "custom_app" and "custom_app_arguments" parameters
custom_app	Path to the client program executable to be used, located on the client machine. Relevant only if the "use_custom_app" option is enabled
custom_app_arguments	Arguments to pass to the client program when launching the application. Relevant only if the "use_custom_app" option is enabled. A number of variables can be specified in these arguments: "%IP%", "%PORT%", "%USER%", "%PASSWORD%", "%RESOURCENAME%" and "%SHELL%".
recorder_timeout_sec	Timeout period for the **cyber**elements Cleanroom session in seconds, in case the recorder fails to start. A value of 0 completely disables the timeout.

Parameter	Description
broker_collection	RDS broker server collection. Applicable only if the "noagent" option is enabled and the "disable_kerberos" and "gw_on_workstation" options are disabled
enable_recording_marker	Enabling the display of a recording indicator to the user in the application window if the session is actually being recorded by **cyber**elements Cleanroom. Applicable only if the "noagent" option is disabled
use_rds_broker	Enabling the use of an RDS broker. Applicable only if the "noagent" option is disabled

Example of creation:

```
data = {
    'resource': {
        'name': 'test rds',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'video_deletable': True,
        'video_storage_duration_days': 365,
        'register_video': True,
        'enable_SSO': 'askPassword',
        'server': '1.2.3.4',
        'socket': 3389,
        'setwindowtitle': True,
    },
    'colors_rdp': 24,
    'resolution': 'fullscreen',
    'use_all_screen': True,
    'enable_CSSP': True,
    'redirect_clipboard': True,
    'mount_local_disk': True,
    'restricted_admin': True,
    'recorder_timeout_sec': 60,
    'enable_recording_marker': False
}

requests.post('https://<mediation_server>/publicapi/<org>/rdpservices/',
json=data, headers={"Authorization":id})
```

2.6.3 Privileged HTML5 RDP applications

URL:

/publicapi/<ORGANIZATION>/html5rdpservices/

/publicapi/<ORGANIZATION>/html5rdpservices/<ID>/

/publicapi/<ORGANIZATION>/html5rdpservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the HTML5 RDP-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:


```
{
    "service_id": 1,
    "resource": {},
    "resolution": "fullscreen",
    "resize_method": "zoom",
    "console_mode": false,
    "colors_html5": 32,
    "remote_application": "",
    "launch_directory": "",
    "auth_domain": "",
    "keyboard": "1036",
    "noagent": false,
    "security_mode": "any",
    "enable_wallpaper": false,
    "forcedomain": false,
    "redirect_html5_clipboard": true,
    "html_clipboard": false,
    "set_tab_title": false,
    "gw_on_workstation": false,
    "gateway_name": "",
    "gateway_zopeid": "",
    "disable_kerberos": false,
    "kerberos_service_accounts": "",
    "rdp_enable_file_transfer": false,
    "recorder_timeout_sec": 30,
    "broker_collection": "",
    "enable_recording_marker": true
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
resolution	Resolution to use for the application window. Can be set to "640x480", "800x600", "1024x768", "1280x1024", "1600x1200", "fullscreen", or "dynamic"
resize_method	Image resizing method. Can mean "zoom" or "display update". Applicable only if the "resolution" parameter is set to "fullscreen" or "dynamic".
console_mode	Enabling *Console* mode for the application
colors_html5	The size of the color palette to be used for the application, in bits. Can be set to 16 or 32. If the "noagent" parameter is disabled, the value 8 is also accepted
remote_application	Name of an executable file on the remote computer to be run automatically when a session is opened
launch_directory	Working directory where the executable specified in "remote_application" is located
auth_domain	Authentication domain to use for authentication on the remote workstation. Applies only if the "forcedomain" option is disabled
keyboard	Keyboard layout to be used for sessions. The string represents the decimal value of the LCID code for the language to be used. The following values are supported: • "1036": French • "1033": English (US) • "99998": AZERTY Unicode • "99999": QWERTY Unicode • "2055": German (Switzerland) • "1031": German (Germany) • "4108": French (Switzerland) • "2070": Portuguese *Note: The values "99998" and "99999" are not official LCID codes; they are used by **cyber**elements Cleanroom for generic keyboards.*
noagent	Enabling the *without an agent* mode for this application
security_mode	RDP security mode. If the "noagent" option is enabled, the values "rdp" and "nla" are accepted. If the "noagent" option is disabled, the values "rdp", "nla", "tls", and "any" are accepted.
enable_wallpaper	Enabling the wallpaper for sessions

Parameter	Description
forcedomain	Enabling the use of the user's login account domain for authentication with the application target
redirect_html5_clipboard	Enable clipboard redirection from the client machine to the session. This redirection works only for text content
html_clipboard	Enabling the extended clipboard, which allows for the management of formatted text content
set_tab_title	Enabling overwriting of session tab titles. If this option is enabled, the title is replaced with the application name (the "name" property in the common settings).
gw_on_workstation	Enabling the *built-in Edge Gateway* feature for the application. Must be used in conjunction with the "gateway_name" and "gateway_zopeid" parameters. Incompatible with the "broker_collection" parameter
gateway_name	Name of the embedded Edge Gateway. Must match the name listed in the Edge Gateway's certificate. Relevant and required only if "gw_on_workstation" is enabled
gateway_zopeid	Zope ID of the embedded Edge Gateway. Relevant and required only if "gw_on_workstation" is enabled
disable_kerberos	Disabling Kerberos for the application. Applies only if the "noagent" option is enabled
kerberos_service_accounts	List of service accounts to use for retrieving the Kerberos TGT. Must be specified as a single string, with each account name separated by a comma. Applicable only if the "noagent" option is enabled and the "disable_kerberos" option is disabled.
rdp_enable_file_transfer	Enabling file transfer in the app
recorder_timeout_sec	Timeout period for the **cyber**elements Cleanroom session if the recorder fails to start, in seconds. A value of 0 completely disables the timeout.
broker_collection	RDS broker server collection. Applicable only if the "noagent" option is enabled and the "disable_kerberos" and "gw_on_workstation" options are disabled
enable_recording_marker	Enabling the display of a recording indicator to the user in the application window if the session is actually being recorded by **cyber**elements Cleanroom. Applicable only if the "noagent" option is disabled

Example of creation:

```
data = {
    'resource': {
        'name': 'test rdp html5',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'video_deletable': True,
        'video_storage_duration_days': 365,
        'register_video': True,
        'enable_SSO': 'askPassword',
        'server': '1.2.3.4',
        'socket': 3389,
    },
    'resolution': 'fullscreen',
    'resize_method': 'zoom',
    'colors_html5': 32,
    'keyboard': '1036',
    'security_mode': 'tls',
    'enable_wallpaper': True,
    'redirect_html5_clipboard': True,
    'html_clipboard': True,
    'set_tab_title': True,
    'rdp_enable_file_transfer': True,
    'recorder_timeout_sec': 60,
    'enable_recording_marker': True
}

requests.post('https://<mediation_server>/publicapi/<org>/html5rdpservices/',
json=data, headers={"Authorization":id})
```

2.6.4 Privileged SSH applications

URL:

/publicapi/<ORGANIZATION>/sshservices/

/publicapi/<ORGANIZATION>/sshservices/<ID>/

/publicapi/<ORGANIZATION>/sshservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the SSH-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:

```
{
    "auto_password": [
        {"pattern": "su -", "alias": "alias_name", "index": 0, "is_regex": false}
    ],
    "service_id": 1,
    "resource": {},
    "cmd_at_launch": "",
    "enable_file_transfer": false,
    "redirect_application": "",
    "redirect_params": "",
    "launch_app": false,
    "use_custom_app": false,
    "custom_app": "",
    "custom_app_arguments": "",
    "prompt_pattern": ""
}
```

Parameter	Description
auto_password	Password injection configurations, in the form of a list of objects, each representing a different configuration. Each configuration consists of the following parameters: • "pattern": the trigger pattern for the injection, which **cyber**elements Cleanroom attempts to detect in the session to trigger a password injection • "alias": the alias name of the **cyber**elements Cleanroom vault that will provide the password to be injected. *Note: If the vault is in Keeper mode, this name must be a Keeper UID* • "index": a positive or zero number representing the priority of the trigger pattern. Patterns are evaluated in ascending order of index, and the numbers must be sequential (no duplicates and no missing numbers) *Note: Only one pattern can be triggered per keystroke. If a pattern is triggered, all patterns with lower priority (i.e., with a higher "index") are ignored* • "is_regex": option indicating that the trigger pattern is a regular expression, not a literal text string
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
cmd_at_launch	Command to be executed in the SSH session once the connection is established. Applicable only if the "enable_file_transfer" option is disabled
enable_file_transfer	Enabling *File Transfer* mode. If this option is enabled, the app will use an SFTP connection instead of an SSH connection.
redirect_application	Deprecated parameter
redirect_params	Deprecated parameter
launch_app	Deprecated parameter
use_custom_app	Enabling the use of a specific client program to launch the application. Use in conjunction with the "custom_app" and "custom_app_arguments" parameters
custom_app	Path to the client program executable to be used, located on the client machine. Relevant only if the "use_custom_app" option is enabled

Parameter	Description
custom_app_arguments	Arguments to pass to the client program when launching the application. Relevant only if the "use_custom_app" option is enabled A number of variables can be specified in these arguments: "%IP%", "%PORT%", "%USER%", "%PASSWORD%", "%RESOURCENAME%" and "%SHELL%"
prompt_pattern	SSH session command prompt

Example of creation:

```
data = {
    'resource': {
        'name': 'test ssh',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'enable_SSO': 'sso-fixe',
        'aliases': ['user_alias_name'],
        'server': '1.2.3.4',
        'socket': 22,
    },
    "auto_password": [
        {"pattern": "^su$", "alias": "root_alias_name", "index": 0, "is_regex":
True},
        {"pattern": "sudo -s", "alias": "root_alias_name", "index": 1, "is_regex":
False},
    ],
}

requests.post('https://<mediation_server>/publicapi/<org>/sshservices/',
json=data, headers={"Authorization":id})
```

Example of updating SSH password injection configurations using the *Keeper*-mode vault:

```
data = {
    "auto_password": [
        {'pattern': '^su$', 'alias': 'GeiUk2qWMdEodi-6SfU3Pw', 'index': 0,
'is_regex': True},
        {'pattern': 'sudo -s', 'alias': 'Qj93wfPHm2LedB7cFjKdzg', 'index': 1,
'is_regex': False},
    ]
}

requests.patch('https://<mediation_server>/publicapi/<org>/sshservices/1/',
json=data, headers={"Authorization": id})
```

2.6.5 Privileged HTML5 SSH applications

URL:

/publicapi/<ORGANIZATION>/html5sshservices/

/publicapi/<ORGANIZATION>/html5sshservices/<ID>/

/publicapi/<ORGANIZATION>/html5sshservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the HTML5 SSH-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:

```
{
    "auto_password": [],
    "service_id": 1,
    "resource": {},
    "cmd_at_launch": "",
    "term_type": "linux",
    "enable_file_transfer": false,
    "redirect_html5_clipboard": true,
    "backspace_key": 127,
    "set_tab_title": false,
    "prompt_pattern": ""
}
```

Parameter	Description
auto_password	Password injection configurations, in the form of a list of objects, each representing a different configuration. Each configuration consists of the following parameters: • "pattern": the trigger pattern for the injection, which **cyber**elements Cleanroom attempts to detect in the session to trigger a password injection • "alias": the alias name of the **cyber**elements Cleanroom vault that will provide the password to be injected. *Note: If the vault is in Keeper mode, this name must be a Keeper UID* • "index": a positive or zero number representing the priority of the trigger pattern. Patterns are evaluated in ascending order of index, and the numbers must be sequential (no duplicates and no missing numbers) *Note: Only one pattern can be triggered per keystroke. If a pattern is triggered, all patterns with lower priority (i.e., with a higher "index") are ignored* • "is_regex": option indicating that the trigger pattern is a regular expression, not a literal text string

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
cmd_at_launch	Command to be executed in the SSH session once the connection is established. Applicable only if the "enable_file_transfer" option is disabled
term_type	Type of device to simulate for sessions using this application
enable_file_transfer	Enabling *File Transfer* mode. If this option is enabled, the app will use an SFTP connection instead of an SSH connection.
redirect_html5_clipboard	Enable clipboard redirection from the client machine to the session. This redirection works only for text content
backspace_key	An integer representing the ASCII code of the backspace character to be used for sessions using this application. The values 8 (the "backspace" character) and 127 (the "delete" character) are supported
set_tab_title	Enabling overwriting of session tab titles. If this option is enabled, the title is replaced with the application name (the "name" property in the common settings).
prompt_pattern	SSH session command prompt

Example of creation:

```
data = {
    'resource': {
        'name': 'test h5 ssh',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'enable_SSO': 'askPassword',
        'server': '1.2.3.4',
        'socket': 22,
    },
    "auto_password": [],
    'term_type': 'linux',
    'backspace_key': 127,
    'set_tab_title': True,
}

requests.post('https://<mediation_server>/publicapi/<org>/html5sshservices/',
json=data, headers={"Authorization":id})
```

2.6.6 Privileged VNC applications

URL:

/publicapi/<ORGANIZATION>/vncservices/

/publicapi/<ORGANIZATION>/vncservices/<ID>/

/publicapi/<ORGANIZATION>/vncservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the VNC-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:

```
{
    "service_id": 1,
    "resource": {},
    "custom_mstsc_options": "",
    "disable_mstsc_certificate_verification": false,
    "use_custom_app": false,
    "custom_app": "",
    "custom_app_arguments": "",
    "gw_on_workstation": false,
    "gateway_name": "",
    "gateway_zopeid": ""
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
custom_mstsc_options	Custom settings for the RDP file used to launch the application
disable_mstsc_certificate_verification	Disabling certificate verification on the client machine. Applies only to *Windows* client computers using *mstsc*
use_custom_app	Enable the use of a client program other than *mstsc*. Use in conjunction with the "custom_app" and "custom_app_arguments" parameters
custom_app	Path to the client program executable to be used, located on the client machine. Relevant only if the "use_custom_app" option is enabled

Parameter	Description
custom_app_arguments	Arguments to pass to the client program when launching the application. Relevant only if the "use_custom_app" option is enabled A number of variables can be specified in these arguments: "%IP%", "%PORT%", "%USER%", "%PASSWORD%", "%RESOURCENAME%" and "%SHELL%"
gw_on_workstation	Enabling the *built-in Edge Gateway* feature for the application. Must be used in conjunction with the "gateway_name" and "gateway_zopeid" parameters. Incompatible with the "broker_collection" parameter
gateway_name	Name of the built-in Edge Gateway. Must match the name listed in the Edge Gateway's certificate. Relevant and required only if "gw_on_workstation" is enabled
gateway_zopeid	Zope ID of the embedded Edge Gateway. Relevant and required only if "gw_on_workstation" is enabled

Example of creation:

```
data = {
    'resource': {
        'name': 'test vnc',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'enable_SSO': 'no-sso',
        'server': '1.2.3.4',
        'socket': 5900,
    },
    'custom_mstsc_options': '',
    'disable_mstsc_certificate_verification': True,
    'use_custom_app': False,
    'gw_on_workstation': False,
}

requests.post('https://<mediation_server>/publicapi/<org>/vncservices/',
json=data, headers={"Authorization":id})
```

2.6.7 Privileged HTML5 VNC applications

URL:

/publicapi/<ORGANIZATION>/html5vncservices/

/publicapi/<ORGANIZATION>/html5vncservices/<ID>/

/publicapi/<ORGANIZATION>/html5vncservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the HTML5 VNC-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:

```
{
    "service_id": 1,
    "resource": {},
    "redirect_html5_clipboard": true,
    "set_tab_title": false,
    "gw_on_workstation": false,
    "gateway_name": "",
    "gateway_zopeid": ""
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
redirect_html5_clipboard	Enable clipboard redirection from the client machine to the session. This redirection works only for text content
set_tab_title	Enabling overwriting of session tab titles. If this option is enabled, the title is replaced with the application name (the "name" property in the common settings).
gw_on_workstation	Enabling the *built-in Edge Gateway* feature for the application. Must be used in conjunction with the "gateway_name" and "gateway_zopeid" parameters. Incompatible with the "broker_collection" parameter
gateway_name	Name of the built-in Edge Gateway. Must match the name listed in the Edge Gateway's certificate. Relevant and required only if "gw_on_workstation" is enabled

Parameter	Description
gateway_zopeid	Zope ID of the embedded Edge Gateway. Relevant and required only if "gw_on_workstation" is enabled

Example of creation:

```
data = {
    'resource': {
        'name': 'test h5 vnc',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'enable_SSO': 'no-sso',
        'server': '1.2.3.4',
        'socket': 5900,
    },
    'set_tab_title': True,
    'gw_on_workstation': False,
}

requests.post('https://<mediation_server>/publicapi/<org>/html5vncservices/',
json=data, headers={"Authorization":id})
```

2.6.8 Privileged Web applications

URL:

/publicapi/<ORGANIZATION>/webrecordservices/

/publicapi/<ORGANIZATION>/ webrecordservices /<ID>/

/publicapi/<ORGANIZATION>/ webrecordservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the Web-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:

```
{
    "service_id": 1,
    "resource": {},
    "secure_connection": true,
    "url": "",
    "target_server": "www.example.com",
    "enable_recording_marker": true,
    "sso_form_mode": 2,
    "sso_login_field": "",
    "sso_password_field": "",
    "sso_form_data": "",
    "sso_advanced_data_enabled": false,
    "sso_advanced_data": "",
    "auth_mode": 2,
    "sso_type": "classic",
    "sso_url": "",
    "html_sso": false,
    "use_form_action_attr": false,
    "additionnal_networks": false,
    "injection_conf": ""
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "priv"
secure_connection	Enabling secure connections to the remote web server. If this option is enabled, communications will use the HTTPS protocol. If it is disabled, the HTTP protocol will be used.
url	Path to the remote web application
target_server	Remote web host targeted by the application. Can be a hostname or an IP address, optionally followed by a port number. This field overrides the value of the common "server" parameter (see Common Parameters)
enable_recording_marker	Enabling the display of a recording indicator to the user in the application window if the session is actually being recorded by **cyber**elements Cleanroom. Applicable only if the "noagent" option is disabled
sso_form_mode	Form mode, as part of SSO for HTML forms. Applicable only if the "html_sso" parameter is enabled and if "sso_type" is set to "classic" or "preload"
sso_login_field	The name of the field in the HTML authentication form that should contain the user's account name. Applicable only if the "html_sso" parameter is enabled, "sso_advanced_data_enabled" is disabled, and "sso_type" is set to "classic" or "preload"
sso_password_field	The name of the field in the HTML authentication form that should contain the user's account password. Applicable only if the "html_sso" parameter is enabled, "sso_advanced_data_enabled" is disabled, and "sso_type" is set to "classic" or "preload"
sso_form_data	Additional settings for the HTML authentication form. Applicable only if the "html_sso" parameter is enabled, "sso_advanced_data_enabled" is disabled, and "sso_type" is set to "classic" or "preload"
sso_advanced_data_enabled	Enabling the advanced configuration mode, allowing you to specify HTML form data in a single parameter, "sso_advanced_data". This renders the "sso_login_field", "sso_password_field", and "sso_form_data" parameters obsolete.
sso_advanced_data	Advanced settings for the HTML authentication form. Applicable only if the "html_sso" and "sso_advanced_data_enabled" parameters are

Parameter	Description
	enabled and if "sso_type" is set to "classic" or "preload"
auth_mode	Authentication type. Can be set to one of the following values: • 1 for "Basic Authentication" • 2 for "Automatic Detection"
sso_type	SSO type on a standard HTML form. Can be set to one of the following values: • "classic" for "Classic" mode • "preload" for "Form Preload" mode • "preauth" for "Pre-authentication" mode • "inject" for "Injection" mode Relevant only if the "html_sso" parameter is enabled *Note: It is currently not possible to configure a functional "preauth" SSO type via the REST API.*
sso_url	Path to the HTML authentication form to be preloaded. Relevant only if the "html_sso" parameter is enabled and "sso_type" is set to "preload"
html_sso	Enabling SSO mode on standard HTML forms. Enables the use of the "sso_type" parameter
use_form_action_attr	Enabling automatic detection of the path to the remote web page based on the "action" attribute of the HTML form. Disables the "url" parameter
additionnal_networks	Enabling additional networks. *Note: It is currently not possible to configure these additional networks via the REST API.*
injection_conf	Injection configuration, as generated by the browser plugin designed for this purpose. Relevant only if the "html_sso" parameter is enabled and "sso_type" is set to "inject"

Example of creation:

```
data = {
    'resource': {
        'name': 'test web sso preload',
        'description': '',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'priv',
        'enable_SSO': 'sso',
    },
    'secure_connection': True,
    'url': '',
    'target_server': 'www.example.com',
    'enable_recording_marker': True,
    'sso_form_mode': 2,
    'sso_login_field': 'user_login',
    'sso_password_field': 'user_pwd',
    'sso_type': 'preload',
    'sso_url': 'api/authform.html',
    'html_sso': True,
    'use_form_action_attr': True,
    "sso_form_data": "",
    "sso_advanced_data": "",
    "auth_mode": 2,
}

requests.post('https://<mediation_server>/publicapi/<org>/webrecordservices/',
json=data, headers={"Authorization":id})
```

2.6.9 Standard Web applications

URL:

/publicapi/<ORGANIZATION>/standardwebrecordservices/

/publicapi/<ORGANIZATION>/standardwebrecordservices/<ID>/

/publicapi/<ORGANIZATION>/standardwebrecordservicesbyname/<NAME>/

The data format and accepted parameters are exactly the same as for privileged web applications, with the exception that the "application_type" must be set to 'std' rather than "priv".

2.6.10 Standard generic tunnel applications

URL:

/publicapi/<ORGANIZATION>/portforwardservices/

/publicapi/<ORGANIZATION>/portforwardservices/<ID>/

/publicapi/<ORGANIZATION>/portforwardservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in Generic Tunnel-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

The data format of the application:

```
{
    "service_id": 1,
    "resource": {},
    "redirect_application": "",
    "redirect_params": "",
    "launch_app": false,
    "alert_user": true,
    "redirect_html_sso": false,
    "launch_web_app": false,
    "enable_url_rewriting": false,
    "preserve_host": false,
    "avoid_windows_auth": false,
    "url_app": "",
    "redirect_sso_login": "",
    "redirect_sso_password": "",
    "redirect_sso_extra": "",
    "redirect_sso_method": 2,
    "default_port": 0,
    "redirections": [
        {
            "id_for_resource": 0,
            "redirect_protocol": "ssh",
            "remote_server": "1.1.1.1",
            "remote_port": 22,
            "local_server": "127.0.0.1",
            "desired_local_port": 0
        }
    ]
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "std"
redirect_application	Executable to be launched when the application starts. Applicable only if the "launch_app" parameter is enabled
redirect_params	Parameters for the redirected executable. Applicable only if the "launch_app" parameter is enabled
launch_app	Enable automatic execution of a redirected executable. Also enables the use of SSO mode with this application (see Common parameters). It is not recommended to use this option if "alert_user" or "launch_web_app" is enabled
alert_user	Enabling the notification that alerts the user when the connection is established. We do not recommend using this option if "launch_app" or "launch_web_app" is enabled
redirect_html_sso	Enabling SSO mode on standard HTML forms. Applies only if "launch_web_app" is enabled
launch_web_app	Enable automatic launch via a web browser. Also enables the use of SSO mode with this application (see Common parameters). It is not recommended to use this option if "launch_app" or "alert_user" is enabled. This applies only if at least one port redirection specified in the "redirections" parameter is configured with one of the following protocols: "http", "https" or "lotus-web"
enable_url_rewriting	Enabling URL rewriting. This applies only if at least one port redirection specified in the "redirections" parameter is configured with one of the following protocols: "http", "https" or "lotus-web"
preserve_host	Enabling *Host* header preservation. This applies only if at least one port redirection specified in the "redirections" parameter is configured with one of the following protocols: "http", "https" or "lotus-web"
avoid_windows_auth	Enabling the Windows authentication challenge. This applies only if at least one port redirection specified in the "redirections" parameter is configured with one of the following protocols: "http", "https" or "lotus-web"
url_app	URL to launch automatically. Relevant only if the "launch_web_app" parameter is enabled

Parameter	Description
redirect_sso_login	The name of the field in the HTML authentication form that should contain the user's account name. Relevant only if the "redirect_html_sso" parameter is enabled
redirect_sso_password	The name of the field in the HTML authentication form that should contain the user's account password. Relevant only if the "redirect_html_sso" parameter is enabled
redirect_sso_extra	Additional settings for the HTML authentication form. Relevant only if the "redirect_html_sso" parameter is enabled
redirect_sso_method	Procedure for submitting the HTML authentication form. Can be set to one of the following values: • 1 for the "GET" method • 2 for the "POST" method
default_port	Default remote port for redirects when the generic tunnel application is in dynamic mode (see Common parameters). *Note: It is not possible to configure a standard generic tunnel application in dynamic mode via the REST API*
redirections	List of configurations for this application. Each redirection must follow a specific format, described below

Redirection data format:

```
{
    "id_for_resource": 0,
    "redirect_protocol": "ssh",
    "remote_server": "1.1.1.1",
    "remote_port": 22,
    "local_server": "127.0.0.1",
    "desired_local_port": 0
}
```

Parameter	Description
id_for_resource	Redirect ID (must be unique within the application)
redirect_protocol	Protocol to be redirected. Can be set to one of the following values: • "ftp" • "http" • "https" • "ica" • "imap"

Parameter	Description
	• "impas" • "mapi" • "notes" • "pop" • "pops" • "prox-funk" • "proxy-web" • "rlogin" • "smtp" • "socks" • "ssh" • "telnet" • "tse" • "selligent" • "lotus-web" • "other-tcp" • "other-udp"
remote_server	Redirect target addresses
remote_port	Redirect target port
local_server	Redirect source address
desired_local_port	Redirect source port

Notes:

- *It is not possible to configure a standard generic tunnel application in dynamic mode via the REST API*
- *SSO settings are only relevant if one of the automatic launch options is enabled ("launch_app" or "launch_web_app")*

Example of creating an application without automatic launch:

```
data = {
    'resource': {
        'name': 'test port forward',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'std',
        'server': '',
        'socket': 0,
    },
    'alert_user': True,
    'redirections': [
        {
            'id_for_resource': 0,
            'redirect_protocol': 'ssh',
            'remote_server': '1.1.1.1',
```

```
            'remote_port': 22,
            'local_server': '127.0.0.1',
            'desired_local_port': 50022
        },
        {
            'id_for_resource': 1,
            'redirect_protocol': 'ssh',
            'remote_server': '2.2.2.2',
            'remote_port': 22,
            'local_server': '127.0.0.1',
            'desired_local_port': 50023
        }
    ]
}

requests.post('https://<mediation_server>/publicapi/<org>/portforwardservices/',
json=data, headers={"Authorization":id})
```

Example of creating an app that launches automatically in the browser:

```
data = {
    'resource': {
        'name': 'test port forward',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'std',
        'enable_SSO': 'sso-fixe',
        'aliases': ['example.com admin']
    },
    'redirect_html_sso': True,
    'launch_web_app': True,
    'enable_url_rewriting': False,
    'preserve_host': False,
    'avoid_windows_auth': True,
    'url_app': 'path/to/index.html',
    'redirect_sso_login': 'user_login',
    'redirect_sso_password': 'user_pwd',
    'redirect_sso_extra': '',
    'redirect_sso_method': 2,
    'redirections': [
        {
            'id_for_resource': 0,
            'redirect_protocol': 'https',
            'remote_server': 'example.com',
            'remote_port': 443,
            'local_server': '127.0.0.1',
            'desired_local_port': 443
        }
    ]
}

requests.post('https://<mediation_server>/publicapi/<org>/portforwardservices/',
json=data, headers={"Authorization":id})
```

2.6.11 Standard VPN applications

URL:

/publicapi/<ORGANIZATION>/vpnservices/

/publicapi/<ORGANIZATION>/vpnservices/<ID>/

/publicapi/<ORGANIZATION>/vpnservicesbyname/<NAME>/

Note: The identifier to specify for URLs is the "service_id" found in the VPN-specific application data, while the expected name is the one found in the "name" parameter of the common application data.

Data format:


```
{
    "service_id": 1,
    "resource": {},
    "display_gui": true,
    "mode": "routing",
    "protocol": "tcp",
    "network": "10.10.0.0",
    "netmask": "255.255.255.0",
    "network_begin": null,
    "netmask_begin": null,
    "network_end": null,
    "netmask_end": null,
    "ip_policy": "dhcp",
    "dns_suffix": "",
    "fulltunnel_enabled": false,
    "routes": [
        {
            "ip": "1.1.1.0",
            "netmask": "255.255.255.0"
        },
        {
            "ip": "2.2.2.0",
            "netmask": "255.255.255.0"
        }
    ],
    "allowed_networks_ip": [
        {
            "ip": "*"
        }
    ],
    "forbidden_networks": [
        {
            "ip": "1.1.1.1"
        }
    ],
    "allowed_networks_ip_range": [
        {
            "begin_ip": "1.1.1.1",
```

```
            "end_ip": "2.2.2.2"
        }
    ],
    "allowed_networks_mask": [
        {
            "ip": "1.1.1.0",
            "netmask": "255.255.255.0"
        },
        {
            "ip": "1.1.1.0",
            "netmask": "24"
        }
    ],
    "dns_servers": [
        {
            "dns_server": "lan.local"
        }
    ]
}
```

Parameter	Description
service_id	Service ID (read-only)
resource	An object containing the application's common parameters. Among these parameters, the "application_type" must be set to "std"
display_gui	Enable visual feedback to the user regarding the VPN status
mode	VPN application mode. Only the "routing" value is supported
protocol	Protocol to use for communication. Can be set to "tcp" or "udp"
network	The VPN network's IP address, to be specified along with the "netmask" parameter
netmask	Subnet mask in decimal notation, to be specified along with the "network" parameter
network_begin	Unused parameter
netmask_begin	Unused parameter
network_end	Unused parameter
netmask_end	Unused parameter
ip_policy	IP address allocation strategy for users of the application. Only the value "dhcp" is supported
dns_suffix	VPN DNS suffix

Parameter	Description
fulltunnel_enabled	Enables "full tunneling" mode, which redirects all network traffic through the VPN tunnel. Makes the "routes" setting unnecessary
routes	A list of routes that will be configured on the user's workstation, allowing them to access the various network segments of the remote site. Each route is an object with two keys, "IP" and "netmask," representing the IP address and the subnet mask, respectively. The mask can only be specified in decimal notation. Relevant only if the "fulltunnel_enabled" option is disabled
allowed_networks_ip	List of *authorized "IP" networks* for the VPN. Each item in the list is an object with an "ip" key whose value is either the address to be authorized or the *wildcard* "*"
forbidden_networks	List of *authorized "IP blocked" networks* for the VPN. Each item in the list must follow the same structure as the items that can be specified for "allowed_networks_ip"
allowed_networks_ip_range	List of authorized "IP range" networks for the VPN. Each item in the list is an object with two keys, 'begin_ip' and "end_ip," whose values are IP addresses—the first and last addresses in the range, respectively.
allowed_networks_mask	List of *authorized "Subnet" networks* for the VPN. Each item in the list is an object with two keys, 'ip' and "netmask," representing the subnet address and subnet mask, respectively. The mask can be specified in decimal or CIDR notation.
dns_servers	List of DNS servers to use for domain name resolution. Each item in the list is an object with a "dns_server" key, whose value can be an IP address or a hostname

Example of creation:

```
data = {
    'resource': {
        'name': 'test vpn',
        'category': 'https://<mediation_server>/publicapi/<org>/categories/<id>/',
        'application_type': 'std',
    },
    'display_gui': False,
    'protocol': 'tcp',
    'network': '10.10.0.0',
    'netmask': '255.255.255.0',
    'dns_suffix': 'local',
    'fulltunnel_enabled': False,
    'routes': [{'ip': '10.10.1.0', 'netmask': '255.255.255.0'}, {'ip':
'10.10.2.0', 'netmask': '255.255.255.0'}],
    'allowed_networks_ip': [{'ip': '*'}],
    'forbidden_networks': [],
    'allowed_networks_ip_range': [],
    'allowed_networks_mask': [],
    'dns_servers': [{'dns_server': 'lan.local'}]
}

requests.post('https://<mediation_server>/publicapi/<org>/vpnservices/',
json=data, headers={"Authorization":id})
```

2.7 Access contract

2.7.1 Basic information

The access contract data corresponds to the data available in the administration console. The associated sites, categories, and applications are listed in the "sites," "categories," and "resources" fields, containing URLs to the corresponding API objects.

However, information related to user groups is not included, as these are managed separately (see Managing Groups and Domains in Contracts).

URL:

publicapi/<ORGANIZATION>/accessprofiles/

publicapi/<ORGANIZATION>/accessprofiles/<ID>/

publicapi/<ORGANIZATION>/accessprofilesbyname/<NAME>/

Data format:

```
{
    "access_profile_id": 1,
    "name": "test access profile",
    "description": "",
    "resources": [
        "https://<mediation_server>/publicapi/<org>/resources/524/"
    ],
    "categories": [
        "https://<mediation_server>/publicapi/<org>/categories/1/",
        "https://<mediation_server>/publicapi/<org>/categories/2/"
    ],
    "sites": [
        "https://<mediation_server>/publicapi/<org>/sites/2/"
    ]
}
```

Parameter	Description
access_profile_id	Access contract ID (read-only)
name	Name of the access contract. Must be unique
description	Description of the access contract
resources	List of REST API URLs that explicitly reference the applications covered by this access contract. An application belonging to a category specified in the "categories" parameter does not need to be specified in "resources"
categories	List of REST API URLs referencing the application categories explicitly covered by this access contract. All applications belonging to these categories are indirectly covered by the access contract. These applications therefore do not need to be specified in the "resources" parameter
sites	List of REST API URLs referencing the sites associated with this access contract, specifying which Edge Gateways and HTML5 Edge Gateways will be available for these accesses.

Example of creation:

```
data = {
    'name': 'test access profile',
    'description': '',
    'resources': [],
    'categories': ['https://<mediation_server>/publicapi/<org>/categories/1/'],
    'sites': ['https://<mediation_server>/publicapi/<org>/sites/2/']
}

requests.post('https://<mediation_server>/publicapi/<org>/accessprofiles/',
json=data, headers={"Authorization":id})
```

2.7.2 Groups and authentication domains

The user groups associated with the various contracts are managed via a specific URL that includes the identifier of the relevant access contract.

URL: publicapi/<ORGANIZATION>/accessprofiles/<ID>/groups/

Note: Only the "GET" and "PUT" methods are accepted for this URL. In addition, these changes completely replace the old list of groups associated with an access contract with the new list provided.

Data format:

```
{
    "name": "group_name",
    "domain": "https://<mediation_server>/publicapi/<org>/domains/<id>/"
}
```

Parameter	Description
name	Group name
domain	In the data received in the response to a GET request, this field contains the REST API URL referencing the domain to which the group belongs. When this parameter is specified in a PUT request to define the groups of an access contract, it must contain the domain name as it is registered in **cyber**elements Cleanroom (see Authentication Domains, "name" parameter)

Example of redefining the groups of an access contract:

```
data = [
    {'name':'group_name_1','domain':'domain_name_1'},
    {'name':'group_name_2','domain':'domain_name_1'},
]

requests.post('https://<mediation_server>/publicapi/<org>/accessprofiles/<id>',
json=data, headers={"Authorization":id})
```

2.7.3 Limitations

Currently, it is not possible to configure generic user groups via the REST API (such as "Everyone" groups, which allow a contract to apply to all groups in a domain without distinction).

It is also not possible to manage access conditions, application restrictions, alerts, network connections, or elements related to personal aliases. All of these elements require the administration console.

2.8 Authentication domains

2.8.1 All types of domains

URL:

publicapi/<ORGANIZATION>/domains/

publicapi/<ORGANIZATION>/domains/<ID>/

publicapi/<ORGANIZATION>/domainbyname/<NAME>/

Note: Only the "GET" method is accepted for these URLs.

Data format:

```
{
    "domain_id": 1,
    "name": "local",
    "cascade_index": -1,
    "microsoft_domain": "",
    "description": "local authentication",
    "deleteOldSession": false
}
```

Parameter	Description
domain_id	Domain ID (read-only)
name	Domain name. Must be unique
cascade_index	Domain priority order in serial authentication. The lower the number, the higher the domain's priority. If the index is -1, the domain is not part of the serial authentication
microsoft_domain	The name of the Microsoft domain that this domain represents in **cyber**elements Cleanroom
description	Description of the access contract
deleteOldSession	Enable automatic deletion of the oldest connection to the user portal when a user reaches their maximum number of concurrent connections

Example of domain retrieval:

```
requests.get('https://<mediation_server>/publicapi/<org>/domains/',
headers={"Authorization":id})
```

Response content:


```
[
    {
        "domain_id": 1,
        "name": "local",
        "cascade_index": -1,
        "microsoft_domain": "",
        "description": "local authentication",
        "deleteOldSession": false
    },
    {
        "domain_id": 2,
        "name": "__guests__",
        "cascade_index": -1,
        "microsoft_domain": "",
        "description": "",
        "deleteOldSession": false
    },
    {
        "domain_id": 3,
        "name": "LDAP Dom 1",
        "cascade_index": -1,
        "microsoft_domain": "systancia.com",
        "description": "",
        "deleteOldSession": false
    }
]
```


2.8.2 Local domains

It is not possible to manage local domains via the REST API.

2.8.3 SAML domains

2.8.3.1 Domain management

URL:

publicapi/<ORGANIZATION>/samldomains/

publicapi/<ORGANIZATION>/samldomains/<ID>/

publicapi/<ORGANIZATION>/samldomainbyname/<NAME>/

Data format:

```
{
    "saml_domain_id": 1,
    "domain": {
        "domain_id": 5,
        "name": "test SAML domain",
        "cascade_index": -1,
        "microsoft_domain": "lan.local",
        "description": "",
        "deleteOldSession": false
    },
    "expiration_minutes": 10,
    "groups_attr": "group-attr",
    "user_attr": "user-attr",
    "email_attr": "",
    "max_connections": 10,
    "IDP": "other",
    "IDP_entityId": "entityId",
    "last_group_update_date": null
}
```

Parameter	Description
saml_domain_id	SAML domain identifier (read-only). This identifier must be included in the URL of requests to the REST API targeting a SAML domain.
domain	An object containing the domain's basic settings. The data format for this object is specified in the "All types of domains" section.
expiration_minutes	Timeout duration for sessions opened by users belonging to the SAML domain, in minutes
groups_attr	Name of the SAML mapping attribute containing the user's group name
user_attr	Name of the SAML mapping attribute containing the user's name
email_attr	The name of the SAML mapping attribute containing the user's email address. Unused parameter by **cyber**elements Cleanroom

Parameter	Description
max_connections	Maximum number of simultaneous connections allowed for this domain
IDP	Type of SAML domain identity provider. Can be set to “azure” or “other”
IDP_entityId	*EntityID* of the SAML domain's identity provider
last_group_update_date	Date of the last synchronization of the SAML domain's Azure groups (read-only). Is updated only for SAML domains with an Azure identity provider

Example of retrieving SAML domains:

```
requests.get('https://<mediation_server>/publicapi/<org>/samldomains/',
headers={"Authorization":id})
```

Response content:

```
[
    {
        'saml_domain_id': 2,
        'domain': {'domain_id': 6, 'name': 'test Azure SAML domain',
'cascade_index': -1, 'microsoft_domain': 'lan.local', 'description': '',
'deleteOldSession': False},
        'expiration_minutes': 10,
        'groups_attr': 'group-attr',
        'user_attr': 'user-attr',
        'email_attr': '',
        'max_connections': 10,
        'IDP': 'azure',
        'IDP_entityId': 'https://sts.windows.net/ID/',
        'last_group_update_date': None
    },
    {
        'saml_domain_id': 1,
        'domain': {'domain_id': 5, 'name': 'test SAML domain', 'cascade_index': -
1, 'microsoft_domain': 'lan.local', 'description': '', 'deleteOldSession': False},
        'expiration_minutes': 10,
        'groups_attr': 'group-attr',
        'user_attr': 'user-attr',
        'email_attr': '',
        'max_connections': 10,
        'IDP': 'other',
        'IDP_entityId': 'entityId',
        'last_group_update_date': None
    }
]
```

Example of SAML domain creation:

```
data = {
    'domain': {
        'name': 'test Azure SAML domain',
        'cascade_index': -1,
        'microsoft_domain': 'lan.local',
        'description': 'Created with REST API',
        'deleteOldSession': False
    },
    'expiration_minutes': 10,
    'groups_attr': 'group-attr',
    'user_attr': 'user-attr',
    'max_connections': 10,
    'IDP': 'azure',
    'IDP_entityId': 'https://sts.windows.net/ID/',
    'email_attr': ''
}

requests.post('https://<mediation_server>/publicapi/<org>/samldomains/',
json=data, headers={"Authorization":id})
```

Example of SAML domain deletion:

```
requests.delete('https://<mediation_server>/publicapi/<org>/samldomains/2/',
headers={"Authorization":id})
```

2.8.3.2 Group management

URL:

publicapi/<ORGANIZATION>/samldomains/<ID_DOMAIN>/groups/

publicapi/<ORGANIZATION>/samldomains/<ID_DOMAIN>/groups/<ID_GROUP>/

Note: The second URL mentioned above does not support any methods other than DELETE.

Data format:

```
{
    "group_id": 1,
    "name": "saml_grp1",
    "external_id": "saml_grp1_ext",
    "description": ""
}
```

Parameter	Description
group_id	SAML group ID (read-only)
name	SAML group name. Must be unique within the SAML domain
external_id	External ID of the SAML group. Used for domains with an Azure identity provider, where it must match the group's ID within Azure
description	Description of the SAML group

Example of retrieving groups from a SAML domain:

```
requests.get('https://<mediation_server>/publicapi/<org>/samldomains/1/groups/',
headers={"Authorization":id})
```

Response content:

```
[
    {'group_id': 1, 'name': 'saml_grp1', 'external_id': 'saml_grp1_ext',
'description': ''},
    {'group_id': 3, 'name': 'saml_grp2', 'external_id': 'saml_grp2_ext',
'description': ''}
]
```

Example of a group creation:

```
data = {
    'name':'saml_grp3',
    'external_id': 'aaaaaaaa-aaaa-aaaa-aaaaaaaaaaaaaaaa',
    'description': 'Created through REST API'
}
```

```
requests.post('https://<mediation_server>/publicapi/<org>/samldomains/1/groups/',
json=data, headers={"Authorization":id})

Response content:
{
    'group_id': 7,
    'name': 'saml_grp3',
    'external_id': 'aaaaaaaa-aaaa-aaaa-aaaaaaaaaaaaaaaa',
    'description': 'Created through REST API'
}
```

Example of a group deletion:

```
requests.delete('https://<mediation_server>/publicapi/<org>/samldomains/1/groups/7
/', headers={"Authorization":id})
```

2.8.4 Admin groups

URL:

publicapi/<ORGANIZATION>/admingroups/

publicapi/<ORGANIZATION>/admingroups/<ID>/

Data format:

```
{
    "admin_group_id": 1,
    "group": "admin group 1",
    "domain": "https://<mediation_server>/publicapi/<org>/domains/<id>",
    "delegated_administrators": false
}
```

Parameter	Description
admin_group_id	Administrator group ID (read-only)
group	Domain name to be defined as an administrator group
domain	A reference to the LDAP or SAML domain to which the group belongs, in the form of a REST API URL
delegated_administrators	Enabling the Delegated administrators group mode

Example of creating an administrators group:

```
data = {
    'group':'ldap_grp_name',
    'domain':'https://<mediation_server>/publicapi/<org>/domains/3/',
    'delegated_administrators':False
}

requests.post('https://<mediation_server>/publicapi/<org>/admingroups/', json=data
headers={"Authorization":id})
```

Response content:

```
{
    'admin_group_id': 3,
    'group': 'ldap_grp_name',
    'domain':'https://<mediation_server>/publicapi/<org>/domains/3/',
    'delegated_administrators': False
}
```

Example of how to change the name of this administrator group:

```
data = {'group': 'ldap_grp_name modified'}

requests.patch('https://<mediation_server>/publicapi/<org>/admingroups/3/',
json=data headers={"Authorization":id})
```

Response content:

```
{
    'admin_group_id': 3,
    'group': 'ldap_grp_name modified',
    'domain': 'https://10.68.243.14/publicapi/<org>/domains/3/',
    'delegated_administrators': False
}
```

2.9 Vault

The REST API only supports the management of vault aliases. It does not support the management of *dynamic aliases*, *password policies*, *supervised accounts*, or *exposure history*.

Additionally, the REST API does not support alias management when the vault is in *Keeper* mode.

Finally, it is not possible to reveal an alias's password or trigger its rotation via the REST API.

2.9.1 Alias of the vault

URL:

/publicapi/<ORGANIZATION>/alias/

/publicapi/<ORGANIZATION>/alias/<ID>/

/publicapi/<ORGANIZATION>/aliasbyname/<NAME>/

Data format:

```
{
    "id": 2,
    "name": "test_alias_name",
    "user_name": "ssh_user",
    "user_domain": "",
    "alias_type": 2,
    "policy": "DefaultPolicy",
    "password": ""
}
```

Parameter	Description
id	Alias ID (read-only)
name	Alias name. Must be unique. Cannot be changed once the alias has been created
user_name	Login for the account represented by the alias
user_domain	The domain name to which the account represented by the alias belongs. Relevant only if the "alias_type" parameter is set to 3
alias_type	Alias type. Can be set to one of the following values: • 1: The alias represents an SSH key • 2: The alias represents an SSH user account (without a domain) • 3: The alias represents an LDAP user account (with a domain)

Parameter	Description
policy	Name of the password policy to which the alias belongs
password	Password or SSH key to be stored in the alias, depending on the "alias_type" setting (write-only). This field is required when creating new aliases. For a type 1 alias, the SSH key must be the private key and must be specified in plain text, without encryption

Example of creating an alias:

```
data = {
    'name': 'test_alias_name',
    'user_name': 'test_username',
    'user_domain': '',
    'alias_type': 2,
    'policy': 'DefaultPolicy',
    'password': 'secret',
}

requests.post('https://<mediation_server>/publicapi/<org>/alias/', json=data,
headers={"Authorization": id})
```

Example of response content:

```
{'id': 8}
```

Example of retrieving data from this alias:

```
requests.get('https://<mediation_server>/publicapi/<org>/alias/8/',
headers={"Authorization": id})
```

Response content:

```
{
    'id': 8,
    'name': 'test_alias_name',
    'user_name': 'test_username',
    'user_domain': '',
    'alias_type': 2,
    'policy': 'DefaultPolicy'
}
```

2.9.2 Alternatives

In addition to the URL described above, vault aliases can be managed via a number of alternative URLs, using information other than the alias ID.

Alternative URLs:

/publicapi/<ORGANIZATION>/aliasbyresourcename/<APPLICATIONNAME>/

/publicapi/<ORGANIZATION>/aliasbylogin/<LOGINALIAS>/

These URLs require additional information, which is used in place of the alias ID to target the alias:

- “aliasbyresourcename” uses the name of the application associated with the alias (see Common parameters, “name” parameter)
- “aliasbylogin” takes the login of the account represented by the alias (see Vault aliases, “user_name” parameter)

These URLs function similarly to the URL /publicapi/<ORGANIZATION>/alias/<ID>/, and notably share the same limitation regarding potential duplicates: if the specified information does not allow for a unique alias to be identified, the request will fail with an http 500 error.

You must therefore be careful when using these alternatives, as they rely on information that can naturally lead to multiple aliases:

- For “aliasbyresourcename,” an application may reference multiple aliases, resulting in a failure.
- For “aliasbylogin”, it is possible to have multiple aliases with the same “user_name” parameter, which also results in a failure. Additionally, the “alias_type” and “user_domain” parameters are not taken into account for these requests.

3 System console API

3.1 Authentication

Every API call must be authenticated. Authentication is based on the "Authorization" HTTP header, which must be included in all requests to the **cyber**elements Cleanroom API. This header must contain a token obtained beforehand by calling a dedicated function at the publicapi/su-api-auth URL.

For the System console API, the username to specify is always "su", while the password is the one used for authentication on the System console.

Example of obtaining the token:

```
import requests

r = requests.post('https://<mediation_server>/publicapi/su-api-auth', json={
    'login': 'su',
    'password': 'secret'
})
data = r.json()
try:
    id = data['id']
    print("Authentication succeeded")
except KeyError:
    print("Authentication failed")
    sys.exit(1)
```

The request must include the following parameters, using JSON syntax:

Parameters	Description
login	The name of the system administrator account. Must always be "su"
password	The system administrator's password

The response must contain a JSON object with an "id" field that contains the expected token. This token must then be included in the "Authorization" header for subsequent requests. The examples in the following sections will include this header.

If the response does not contain an "id" field, authentication has failed.

3.2 Reloading the Apache service

URL:

/publicapi/apache/reload/

Note: This URL only accepts POST requests with no body content.

Modifying elements of the system console via this REST API may affect the configuration of the Apache service on Mediation Controller servers. Therefore, changes may require the service to be reloaded, which does not occur automatically.

You can manually trigger this operation from the administration console using the URL above.

3.3 Organizations

URL:

/publicapi/organizations/

/publicapi/organizations/<ID_OR_NAME>/

/publicapi/organizations/<ID_OR_NAME>/delete/

Note: This URL only accepts POST requests and is equivalent to a DELETE request using the second URL.

Organization operations accept different parameters depending on whether the API request is an organization creation request or not. Below is the structure of the base data, which corresponds to what is expected for a modification (PUT or PATCH) and what will be returned as a response to read requests.

Basic data format:

```
{
    "org_id": 1,
    "name": "organization_1",
    "admin_pwd": "",
    "allow_html5": true,
    "allow_acm": false,
    "su_organization_ip_set": [
        "1.1.1.1",
        "1.1.1.2",
        "1.1.1.3"
    ],
    "creation_state": "2",
    "safe_max_user": 5,
    "nb_max_sessions_user": 2
}
```

Parameters	Description
org_id	Organization ID (read-only)
name	Name of the organization
admin_pwd	Default administrator password for the organization's "*local*" domain (write-only)
allow_html5	Enabling HTML5 connections for this organization, allowing the use of Edge Gateway and HTML5 applications
allow_acm	Enabling the *Application Credential Manager* (*ACM*) for this organization
su_organization_ip_set	List of IP addresses authorized to connect to the organization's administration console (read-only)
creation_state	Current status of the organization (read-only). Can be set to one of the following values: • "-1": Imminent creation • "0": Creation interrupted due to an error • "1": Currently being created • "2": Successfully created • "3": Waiting for the specified Edge Gateway to connect *Note: Although this status is represented by an integer, the value is transmitted as a string*
safe_max_user	Maximum number of user sessions
nb_max_sessions_user	Maximum number of concurrent sessions per user

When creating an organization, a number of additional fields are required, including those related to the database server that hosts or will host the organization database. These parameters are not included in the data returned for GET requests and cannot be edited.

Creating an organization is an asynchronous task, and the creation request will receive a response before the organization is actually created successfully. To reflect this, a creation request receives a 202 "Accepted" status code upon success, instead of the usual 201 "Created" code. Additionally, this response has no content. Once this response is received, the progress of the creation can be tracked via the "creation_state" parameter, which is included in the organization data returned in response to GET requests.

It is currently not possible to specify an Edge Gateway to be used for organization creation. This feature is only available via the **cyber**elements Cleanroom system console.

Data format to be transmitted for creation only:

```
{
    "name": "ipdivasafe",
    "db_login": "database_user",
    "db_pwd": "secret",
    "db_host": "127.0.0.1",
    "db_type": "PostgreSQL",
    "db_ssl_mode": "verify-full",
    "db_pkiid": "1750162818913",
    "db_caid": "2411074302",
    "admin_pwd": "secret2",
    "db_port": 5432,
    "allow_html5": true,
    "allow_acm": true,
    "su_organization_ip_set": ["1.1.1.1", "1.1.1.2", "1.1.1.3"],
    "create_database": true,
    "safe_max_user": 5,
    "nb_max_sessions_user": 2
}
```

Parameters	Description
name	Name of the organization
db_login	Username to use to access the organization's database
db_pwd	The account password to use to access the organization's database
db_host	Address of the server hosting the organization's database
db_type	Database server type. Can be “PostgreSQL” or “Microsoft SQL Server”
db_ssl_mode	SSL Use Policy. Can be set to one of the following values: • “prefer”: Prefer SSL without verifying the certificate (not secured) • “verify-ca”: Verify the server certificate • “verify-full”: Check the server certificate and its name
db_pkiid	The PKI identifier to use (see PKI, “id” parameter). Applicable only if the “db_ssl_mode” parameter is set to “verify-ca” or “verify-full”
db_caid	The ID of the PKI certificate authority to be used (see Certificate Authorities, “id” parameter). Applicable only if the “db_ssl_mode” parameter is set to “verify-ca” or “verify-full”
admin_pwd	Default administrator password for the organization's “*local*” domain
db_port	Port to use for communication with the organization's database
allow_html5	Enable HTML5 connection authorization, allowing the use of Edge Gateways and HTML5 applications

Parameters	Description
allow_acm	Enabling the *Application Credential Manager* (*ACM*) for this organization
create_database	Enable database creation. If this option is enabled, the organization's database will be automatically created on the specified database server. If it is disabled, the organization's database must already exist.
su_organization_ip_set	List of IP addresses authorized to connect to the organization's administration console
safe_max_user	Maximum number of user sessions
nb_max_sessions_user	Maximum number of concurrent sessions per user

Example of creation:

```
data = {
    'name': 'dummy_2',
    'db_login': 'database_user',
    'db_pwd': 'secret',
    'db_host': '127.0.0.1',
    'db_type': 'PostgreSQL',
    'db_ssl_mode': 'prefer',
    'admin_pwd': 'secret2',
    'db_port': 5432,
    'allow_html5': True,
    'allow_acm': True,
    'create_database': False,
    'safe_max_user': 10,
    'nb_max_sessions_user': 1,
    'su_organization_ip_set': ['1.1.1.1', '1.1.1.2', '1.1.1.3']
}

requests.post('https://<mediation_server>/publicapi/organizations/', json=data,
headers={"Authorization": id})
```

Example of checking the progress of the organization's creation:

```
response = requests.get('https://<mediation_server>/publicapi/organizations/4/',
headers={"Authorization": id})
try:
    creation_state = response.json()['creation_state']

    creation_ongoing = creation_state in ['-1', '1', '3']
    created_successful = creation_state == '2'
    creation_failed = creation_state == '0'
except requests.exceptions.JSONDecodeError | KeyError:
    # Request failed
    pass
```

3.4 PKI and related data

3.4.1 Public Key Infrastructure (PKI)

URL:

/publicapi/pki/

/publicapi/pki/<NAME>/

Note: It is not possible to create, modify, or delete PKIs via the REST API. Only GET requests are accepted. All parameters listed below are therefore read-only.

Data format:

```
{
    "name": "PKI name",
    "description": "",
    "usage": "ABC",
    "active": false,
    "id": "pki_id",
    "casDir": "/etc/ipdiva/pkis/pki_id/ca",
    "hasCa": true,
    "hasCert": true
}
```

Parameters	Description
name	Name of the PKI. Must be unique
description	Description of the PKI
usage	How to use PKI. Can be set to one of the following values: • "A": Certificates for web servers • "B": Certificates for user authentication • "C": Certificates for Edge Gateways/Mediation Controller servers • "AB": Certificates for web servers and user authentication • "AC": Certificates for web servers and Edge Gateways/Mediation Controller servers • "BC": Certificates for Edge Gateways/Mediation Controller servers and user authentication • "ABC": Certificates for web servers, user authentication, and Edge Gateways/Mediation Controller servers
active	Unused parameter
id	Unique PKI identifier
casDir	Path to the location, on the Mediation Controller server, of the data related to the certification authorities for this PKI.

Parameters	Description
hasCa	Flag indicating whether the PKI has at least one configured certification authority
hasCert	Flag indicating whether at least one of the certification authorities configured for this PKI has a certificate

Example of retrieving a PKI named “PKI test”:

```
requests.get('https://<mediation_server>/publicapi/pki/PKI test/',
headers={"Authorization": id})
```

Response content:

```
{
    'name': 'PKI test',
    'description': '',
    'usage': 'ABC',
    'active': False,
    'id': '1750162818913',
    'casDir': '/etc/ipdiva/pkis/1750162818913/ca',
    'hasCa': True,
    'hasCert': True
}
```

3.4.2 Certification authorities (CA)

URL:

/publicapi/pki/<PKI_NAME>/ca/

/publicapi/pki/<PKI_NAME>/ca/<CA_NAME>/

Note: It is not possible to create, modify, or delete certification authorities via the REST API. Only GET requests are accepted. All parameters listed below are therefore read-only.

Data format:

```
{
    "name": "CA-NAME",
    "id": "id",
    "isRootCA": true,
    "subject": "/CN=CA-NAME",
    "issuer": "/CN=CA-NAME",
    "certFilePath": "/etc/ipdiva/pkis/pki_id/ca/id.pem",
    "hasCert": true
}
```

Parameters	Description
name	Name of the certification authority
id	ID of the certification authority
isRootCA	Flag indicating whether the certification authority is a *root* authority
subject	Subject of the certification authority's certificate
issuer	Issuer of the certificate from the certification authority
certFilePath	Path to the location of the certification authority certificate file on the Mediation Controller server
hasCert	Flag indicating whether the certification authority has at least one registered certificate

Example of retrieving certification authorities from a "PKI test" PKI:

```
requests.get('https://<mediation_server>/publicapi/pki/PKI test/ca/',
headers={"Authorization": id})
```

Response content:

```
[
    {
        'name': 'TEST-CA',
        'id': '2411074302',
        'isRootCA': True,
        'subject': '/CN=TEST-CA',
        'issuer': '/CN=TEST-CA',
        'certFilePath': '/etc/ipdiva/pkis/1750162818913/ca/2411074302.pem',
        'hasCert': True
    }
]
```

3.4.3 Certificates of certification authorities

URL:

/publicapi/pki/<PKI_NAME>/ca/<CA_NAME>/cert/

/publicapi/pki/<PKI_NAME>/ca/<CA_NAME>/cert/<CERT_NAME>/

Note: It is not possible to create, modify, or delete certificates from certification authorities via the REST API. Only GET requests are accepted. All parameters listed below are therefore read-only.

Data format:

```
{
    "name": "cert_name",
    "id": "cert_id",
    "subject": "/CN=subject",
    "issuer": "/CN=issuer",
    "notBefore": "2023-01-01 00:00:01",
    "notAfter": "2025-01-01 00:00:01",
    "certFilePath": "/etc/ipdiva/pkis/pki_id/ca/ca_id/cert_id.crt",
    "keyFilePath": "/etc/ipdiva/pkis/pki_id/ca/ca_id/cert_id.key",
    "kind": {
        "client": false,
        "clientCa": false,
        "server": false,
        "serverCa": false
    }
}
```

Parameters	Description
name	Certificate name
id	Internal ID for the certificate
subject	Subject of the certificate
issuer	Certificate issuer
notBefore	Certificate validity start date
notAfter	Certificate validity end date
certFilePath	Path to the location on the Mediation Controller server where the signed certificate file is stored
keyFilePath	Path to the location on the Mediation Controller server where the file containing the certificate's private key is stored
kind	Object containing information about the possible uses for this certificate, in accordance with the extensions defined in the certificate.

Example of retrieving the list of certificates registered for a "TEST-CA" certification authority associated with a "PKI test" PKI:

```
requests.get('https://<mediation_server>/publicapi/pki/PKI test/ca/TEST-CA/cert/',
headers={"Authorization": id})
```

Response content:

```
[
        {
            'name': '*.lan.local',
            'id': '2628909733',
            'subject': '/CN=*.lan.local',
            'issuer': '/CN=TEST-CA',
            'notBefore': '2025-04-10 09:15:19',
            'notAfter': '2027-04-10 09:15:19',
            'certFilePath':
'/etc/ipdiva/pkis/1750162818913/ca/2411074302/2628909733.crt',
            'keyFilePath':
'/etc/ipdiva/pkis/1750162818913/ca/2411074302/2628909733.key',
            'kind': {'client': False, 'clientCa': False, 'server': True,
'serverCa': False}
        }
]
```

3.5 Virtual hosts

URL:

/publicapi/virtualhosts/

/publicapi/virtualhosts/<ID>/

/publicapi/virtualhostsbyname/<NAME>/

These URLs list all configured virtual hosts, regardless of their respective types. However, the data for virtual hosts varies depending on their type.

3.5.1 Common data

Virtual hosts share a number of common settings, regardless of their type.

Format of common data:

```
{
    "name": "test_virtualhost",
    "domainName": "local",
    "adminMail": "admin@lan.local",
    "specificSSLCert": false,
    "SSLConfigOverride": {},
    "SSLUserAuthConfig": false,
    "SSLUserAuthConfigOverride": {},
    "type": "type",
    "id": "test_virtualhost",
    "forInterface": "type : test_virtualhost"
}
```

Parameters	Description
name	Virtual host name *Note: It is not recommended to use a name that contains one or more spaces*
domainName	Domain name to which the virtual host should apply
adminMail	Administrator's email address
specificSSLCert	Unused parameter
SSLConfigOverride	Unused parameter
SSLUserAuthConfig	Unused parameter
SSLUserAuthConfigOverride	Unused parameter
type	Virtual host type. Can be set to one of the following values: • "webvpn" for a *Web VPN* virtual host

Parameters	Description
	• "reverseproxy" for a *Reverse Proxy* virtual host • "transparentreverseproxy" for a *Transparent Reverse Proxy* virtual host
id	Virtual host ID (read-only). The same as the name given to the virtual host when it was created
forInterface	Unused parameter

Example of virtual host retrieval:

```
requests.get('https://<mediation_server>/publicapi/virtualhosts/',
headers={"Authorization": id})
```

Response content:

```
[
    {
        'name': 'default',
        'domainName': '',
        'adminMail': 'support@ipdiva.com',
        'specificSSLCert': False,
        'SSLConfigOverride': {},
        'SSLUserAuthConfig': False,
        'SSLUserAuthConfigOverride': {},
        'publishPortal': True,
        'baseRedirection': '/gate/cloud/',
        'HTTPRedirect': False,
        'HTTPRedirectDomains': [],
        'enableActiveSync': True,
        'orgActiveSync': '',
        'enableCARE': True,
        'enableShibboleth': False,
        'enableCyberelements': False,
        'filterOrgs': False,
        'filterOrgList': [],
        'setCybeltOrg': '',
        'type': 'webvpn',
        'id': 'default',
        'forInterface': 'webvpn : default'
    },
    {
        'name': 'Test_RP',
        'domainName': 'lan.local',
        'adminMail': 'admin@lan.local',
        'specificSSLCert': False,
        'SSLConfigOverride': {},
        'SSLUserAuthConfig': False,
        'SSLUserAuthConfigOverride': {},
        'authenticationRedirect': 'auth_redirection.lan.local',
        'type': 'reverseproxy',
        'id': 'Test_RP',
        'forInterface': 'reverseproxy : Test_RP'
```

```
    },
    {
        'name': 'Test_RP_Transparent',
        'domainName': 'lan.local',
        'adminMail': 'admin@lan.local',
        'specificSSLCert': False,
        'SSLConfigOverride': {},
        'SSLUserAuthConfig': False,
        'SSLUserAuthConfigOverride': {},
        'orgTransparentRP': 'organization01',
        'targetTransparentRP': 'target.lan.local',
        'protocolTransparentRP': 'https',
        'siteTransparentRP': '1',
        'activeGatewaysTransparentRP': ['test gateway'],
        'passiveGatewaysTransparentRP': [],
        'cacheTransparentRP': True,
        'preserveHostTransparentRP': False,
        'redirectionsTransparentRP': [],
        'ntlmConfig': False,
        'mappingTransparentRP': [],
        'type': 'transparentreverseproxy',
        'id': 'Test_RP_Transparent',
        'forInterface': 'transparentreverseproxy : Test_RP_Transparent'
    }
]
```

3.5.2 Web VPN virtual host data

Data format:

```
{
    "publishPortal": true,
    "baseRedirection": "",
    "HTTPRedirect": false,
    "HTTPRedirectDomains": [
        ""
    ],
    "enableActiveSync": false,
    "orgActiveSync": "",
    "enableCARE": true,
    "enableShibboleth": false,
    "enableCyberelements": false,
    "filterOrgs": false,
    "filterOrgList": [],
    "setCybeltOrg": ""
}
```

Parameters	Description
publishPortal	Enabling the publication of the **cyber**elements Cleanroom portal using this virtual host
baseRedirection	Base URL redirection, used when a user attempts to access the root of the virtual host
HTTPRedirect	Enabling HTTP redirection, automatically redirecting users to the HTTPS interface if they attempt to access the HTTP interface
HTTPRedirectDomains	List of domain names to be subject to HTTP redirection. Applicable only if the "HTTPRedirect" option is enabled
enableActiveSync	Unused parameter
orgActiveSync	Unused parameter
enableCARE	Unused parameter
enableShibboleth	Enabling the SAML module in **cyber**elements Cleanroom, based on *Shibboleth*
enableCyberelements	Unused parameter
filterOrgs	Unused parameter
filterOrgList	Unused parameter
setCybeltOrg	Unused parameter

3.5.3 Reverse Proxy virtual host data

Data format:

```
{
    "authenticationRedirect": "auth_redirection.lan.local"
}
```

Parameters	Description
authenticationRedirect	The name of the host to which users should be redirected for authentication if they are not already authenticated

3.5.4 Transparent Reverse Proxy virtual host data

Data format:

```
{
    "orgTransparentRP": "organization01",
    "targetTransparentRP": "target.lan.local",
    "protocolTransparentRP": "https",
    "siteTransparentRP": "site_id ",
    "activeGatewaysTransparentRP": [
        "test gateway"
    ],
    "passiveGatewaysTransparentRP": [],
    "cacheTransparentRP": true,
    "preserveHostTransparentRP": false,
    "redirectionsTransparentRP": [],
    "ntlmConfig": false,
    "mappingTransparentRP": []
}
```

Parameters	Description
orgTransparentRP	Name of the **cyber**elements Cleanroom organization containing the site to be used to access the target service
targetTransparentRP	Service targeted by the transparent reverse proxy
protocolTransparentRP	Protocol to use to access the target service. Can be "http" or "https"
siteTransparentRP	Site identifier to use when connecting to the target service. Must be part of the organization specified in the "orgTransparentRP" parameter *Note: Although this identifier is an integer, the parameter must be a string representing that number*
activeGatewaysTransparentRP	List of active Edge Gateway names for the site specified in the "siteTransparentRP" parameter (read-only)
passiveGatewaysTransparentRP	List of names of passive Edge Gateways for the site specified in the "siteTransparentRP" parameter (read-only)
cacheTransparentRP	Enabling the *Apache* HTTP cache on the Mediation Controller server
preserveHostTransparentRP	Unused parameter
redirectionsTransparentRP	Unused parameter
ntlmConfig	Unused parameter
mappingTransparentRP	Unused parameter

3.6 Web interfaces

URL:

/publicapi/webinterfaces/

/publicapi/webinterfaces/<NAME>/

Data format:

```
{
    "name": "default",
    "authType": "none",
    "crls": [""],
    "verifyDepth": 2,
    "certId": "",
    "pkiSelectId": "pki_id",
    "caSelectId": "ca_id",
    "certSelectId": "cert_id",
    "authCaIds": [
        "ca_id@pki_id",
        "ca_id@pki_id"
   ],
    "crlIds": [""],
    "enableOCSPStapling": false,
    "vhs": [
        ["default", true]
    ],
    "hostIds": ["standalone"],
    "webInterfaceId": "<ip_address>:<port>",
    "addressId": ""
}
```

Parameters	Description
name	Web interface name (read-only). Must be unique. For web interfaces other than the *default* one, this name corresponds to the interface's address and port.
authType	Certificate-based authentication management method. Can be set to one of the following values: • "none" to disable it • "optional" to make it optional • "require" to make it mandatory
crls	Unused parameter
verifyDepth	Depth of certificate verification
certId	Unused parameter
pkiSelectId	The PKI identifier configured for this web interface (see Public Key Infrastructure (PKI), "id" parameter)

Parameters	Description
caSelectId	The ID of the certification authority configured for this web interface (see Certification authorities (CA), "id" parameter). This authority must be part of the PKI specified via the "pkiSelectId" parameter
certSelectId	The ID of the certificate configured for this web interface (see "Certificates of certification authorities," 'id' parameter). This certificate must belong to the certification authority specified via the "caSelectId" parameter
authCaIds	List of certification authorities authorized to issue user certificates. Each item in this list represents a certification authority in the form of a character string following the syntax below: "ca_id@pki_id"
crlIds	Unused parameter
enableOCSPStapling	Unused parameter
vhs	List of virtual hosts associated with this web interface. Each item in this list represents a virtual host, displayed as a list showing the host's name and its "active" status.
hostIds	List of Mediation Controller servers affected by the web interface. In *standalone* mode, this must always contain a single value: "standalone". In *cluster* mode: • If the Web interface address is the actual address of one of the servers (*master* or *slave*), this list must contain a single value: "master" or "slave", respectively. • Otherwise, this list may contain either one of these values or both.
webInterfaceId	Web interface address and port (creation only, write-only). Must be unique. The value must follow the syntax "<ip_address>:<port>". Once the web interface has been created, its "name" parameter will contain the value specified in this field
addressId	A flag indicating whether the web interface address is specific. Can be set to one of the following values: • "webRip" if the address matches the actual IP address of the *master* or *slave* Mediation Controller server • "webVip" if the address matches the virtual IP address of the *cluster* • "" (empty string) for all other cases, i.e., for a *standalone* installation or for a custom address on a *cluster* installation

```
Example of creating a Web interface:
data = {
    "webInterfaceId": "<mediation_server>:8443",
    "authType": "none",
    "crls": [""],
    "verifyDepth": 2,
    "certId": "",
    "pkiSelectId": "1750162818913",
    "caSelectId": "2411074302",
    "certSelectId": "2628909733",
    "authCaIds": [""],
    "crlIds": [""],
    "enableOCSPStapling": false,
    "vhs": [
        ["default", true]
    ],
    "hostIds": ["standalone"],
    "addressId": ""
}

requests.post('https://<mediation_server>/publicapi/webinterfaces/',
json=data, headers={"Authorization": id})
```