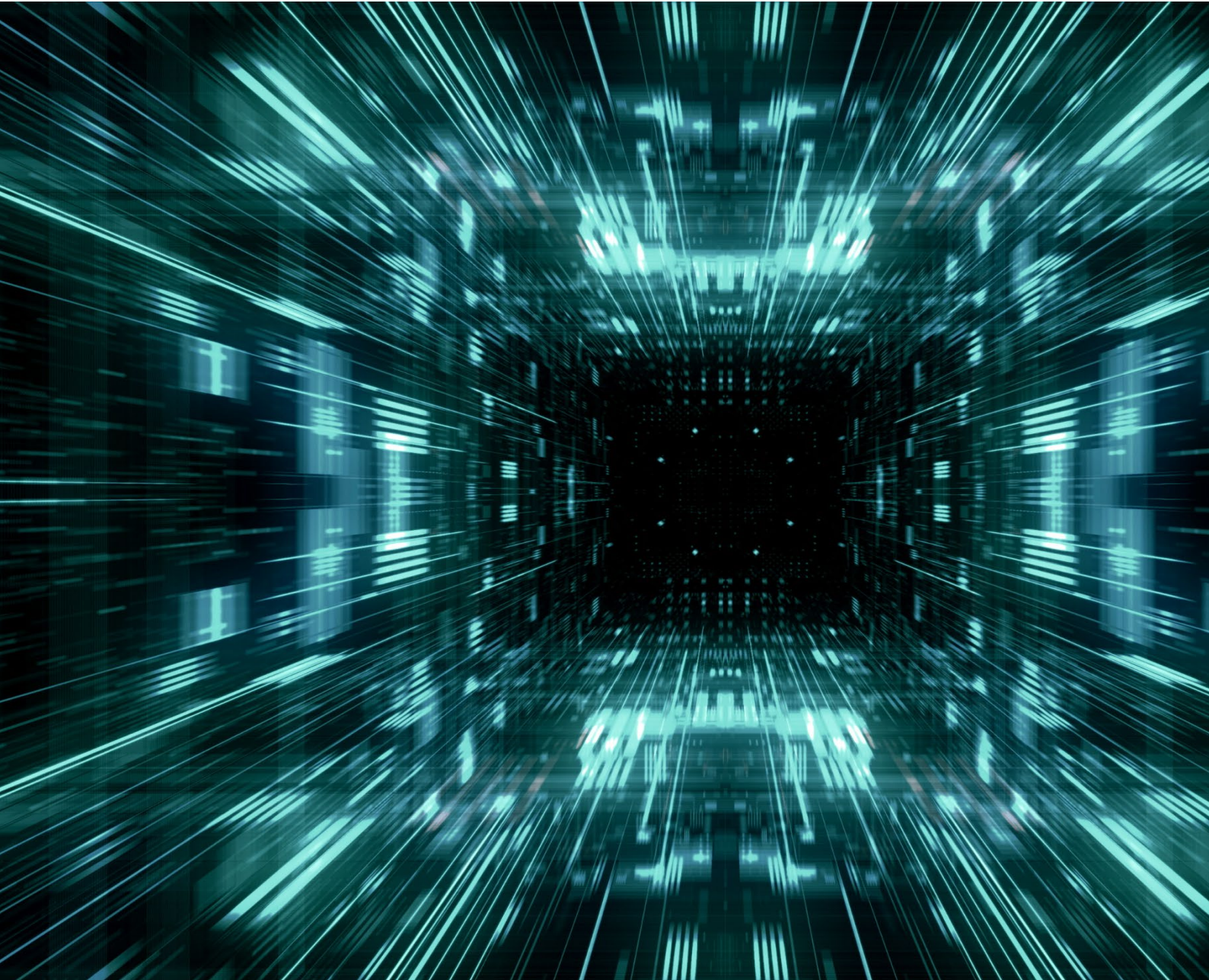


# DAIM Edge Computing Platform API Specification



30.05.2023

No liability is assumed for any damage caused by applying examples contained in this document.

Copyright by DAIM GmbH, All rights reserved

# Edge Device Management API Specification

2.0.8 OAS3

Edge Device Management API endpoints.

EdgeDevices are managed entities that represent a physical device. Each device is owned by a Customer. One or multiple Machines can be attached to an Edge Device. Each EdgeDevice has a dedicated SystemVersion that defines the OS packages, file system structure and management services that are running. Each EdgeDevice has a dedicated Deployment that defines, which Edge Applications Images can be running as Containers on the EdgeDevice.

Each Deployment consists of a set of dedicated Application Images that are tuples of Application and Images that are to be deployed as Edge Applications Containers. Each Application defines a unique container name and docker image that can be running on an Edge Device.

Configurations are the central element of how to configure a running edge application. It can contain either a set of environment variables or ansible variables. ConfigurationFiles are used to mount external files into a running edge application. It can contains a path, filename and the file content. Configurations and ConfigurationFiles are always associated with an distinct Application via the property applicationId. By added tags to Configurations and ConfigurationFiles it is possible to control, on which EdgeDevices they will be applied when a Deployment is triggered. If an EdgeDevice contains at least on tag that is also present in a Configuration or ConfigurationFile, it will be applied to the corresponding Application when it is part of the EdgeDevice`s Deployment. ENV variables will be set when container is started. Configured files will be created and mounted inside docker container on startup. Variables are used by edgedevice-update (script executed when deployment is triggered) and can be set in Applications and Configurations, which will have affects on deployed containers on the device. Following variables are allowed to be set:

- **aliases**: A list of aliases for this container in the network apps\_network. Always contains the app name as element. Examples:
  - "aliases": "[\"database\"]"
- **command**: Command to execute when the container starts. A command may be either a string or a list. Overrides the command defined by the container image. Examples:
  - "command": "[\"supervisord\", \"-nc\", \"/etc/supervisor/supervisord.conf\"]"
  - "command": "/run.sh"
- **data\_target**: Path in container where to bind an app-specific data volume. This automatically creates a docker volume {{ app }}\_data if set. Examples:
  - "data\_target": "/data"
- **extra\_hosts**: Dictionary of host-to-IP mappings to be added in container's /etc/hosts. Always contains entries host.docker.internal and gateway.docker.internal for the docker host IP address. Examples:
  - "extra\_hosts": "{\"example.net\": \"192.0.2.42\"}"
- **memory**: The memory limit in format <number><unit>. Number is a positive integer. Unit can be B (byte), K (kibibyte, 1024B), M (mebibyte), G (gibibyte), T (tebibyte), or P (pebibyte). Omitting the unit defaults to bytes. Examples:
  - "memory": "256M"
- **network\_mode**: Connect the container to a network. Only set to "host" if absolutely required. Choices: bridge (= default), host. Examples:
  - "network\_mode": "host"
- **ports**: List of ports to publish from the container to the host. Examples:
  - "ports": "[\"8080:80\", \"120.0.0.1:9080:81\", \"9000-9005:8001-8006\", \"0.0.0.0:8443:443\"]"
- **tmpfs**: List of tmpfs directories. Examples:
  - "tmpfs": "[\"/run\"]"
- **workdir**: Path to the working directory. Overrides the workdir defined by the container image. Examples:
  - "workdir": "/home/app"

No other edgedevice-update variables can be set via the Edge Device Management API (whitelist). Each Machine represents a physical machine at the Customer site that can be connected to an Edge Device.

Machines can be grouped in Sites, each Machine can only be assigned to one Site or no Site. Machines can be attached to either exactly one Edge Device or no Edge Device.

[Terms of service](#)

[DAIM GmbH - Website](#)

[Send email to DAIM GmbH](#)

[Commercial License](#)

### Servers

<http://localhost:9010/api/v1> - DAIM Device Management API

## Edge Devices



<b>POST</b>	<code>/edgeDevices</code>	Post a single new edge device object	<code>createEdgeDevice</code>	▼
<b>GET</b>	<code>/edgeDevices</code>	Fetch all existing edge devices	<code>getAllEdgeDevices</code>	▼
<b>GET</b>	<code>/edgeDevices/profiles</code>	Fetch all existing edge devices profiles	<code>getAllEdgeDeviceProfiles</code>	▼
<b>POST</b>	<code>/edgeDevices/bulkDeployments</code>	Perform bulk Deployment which is currently associated to given list of EdgeDevices (Note: this operation cannot be undone)	<code>performBulkDeployment</code>	▼
<b>GET</b>	<code>/edgeDevices/bulkDeployments/{uuid}</code>	Get status of running bulk Deployment by given uuid that was started via POST to <code>/edgeDevices/bulkDeployments</code>	<code>getBulkDeploymentStatus</code>	▼
<b>GET</b>	<code>/edgeDevices/search/findByForeignId</code>	Find EdgeDevices by Foreign ID	<code>findEdgeDevicesByForeignId</code>	▼
<b>GET</b>	<code>/edgeDevices/{id}</code>	Fetch single edge devices with given id	<code>getEdgeDevice</code>	▼
<b>PUT</b>	<code>/edgeDevices/{id}</code>	Update Edge Device	<code>putEdgeDevice</code>	▼
<b>PATCH</b>	<code>/edgeDevices/{id}</code>	Update Edge Device	<code>patchEdgeDevice</code>	▼
<b>DELETE</b>	<code>/edgeDevices/{id}</code>	Delete single edge device with given id	<code>deleteEdgeDevice</code>	▼
<b>GET</b>	<code>/edgeDevices/{id}/customer</code>	Get Customer associated to EdgeDevice	<code>getCustomerOfEdgeDevice</code>	▼
<b>GET</b>	<code>/edgeDevices/{id}/containers</code>	Fetch containers of a single edge devices with given id	<code>getContainersOfEdgeDevice</code>	▼

<b>GET</b>	/edgeDevices/{id}/containers/{name}	Fetch single container with given name of a single edge devices with given id	<b>getContainerByNameOfEdgeDevice</b>	✓
<b>GET</b>	/edgeDevices/{id}/machines	Fetch all attached machines of a single edge devices with given id	<b>getMachinesOfEdgeDevice</b>	✓
<b>POST</b>	/edgeDevices/{id}/machines	Attach machine to a single edge devices with given id	<b>createMachineForEdgeDevice</b>	✓
<b>GET</b>	/edgeDevices/{id}/configurations	Find all Configurations whose tag is equal to one of the tags in this EdgeDevice's tags list.	<b>getConfigurationsOfEdgeDevice</b>	✓
<b>POST</b>	/edgeDevices/{id}/configurations	Add new configuration to a single edge device with given id	<b>createConfigurationForEdgeDevice</b>	✓
<b>GET</b>	/edgeDevices/{id}/configurationFiles	Find all ConfigurationFiles whose tag is equal to one of the tags in this EdgeDevice's tags list.	<b>getConfigurationFilesOfEdgeDevice</b>	✓
<b>POST</b>	/edgeDevices/{id}/configurationFiles	Add new configuration files to a single edge device with given id	<b>createConfigurationFileForEdgeDevice</b>	✓
<b>GET</b>	/edgeDevices/{id}/deployment	Fetch active deployment of a single edge device with given id	<b>getDeploymentOfEdgeDevice</b>	✓
<b>POST</b>	/edgeDevices/{id}/deploy	Perform Deployment which is currently associated to this EdgeDevice (optional query parameter "dryRun" to get a preview of the configuration which would be applied, when a new Deployment on this EdgeDevice is performed)	<b>performDeployment</b>	✓
<b>GET</b>	/edgeDevices/{id}/gitLogs	Fetch git logs of a single edge device with given id	<b>getGitLogsOfEdgeDevice</b>	✓
<b>POST</b>	/edgeDevices/{id}/logs	Trigger recording of system logs; the returned UUID can be used to download them using the GET method as soon as they are available	<b>triggerEdgeDeviceLogRecording</b>	✓
<b>GET</b>	/edgeDevices/{id}/logs/{uuid}	Fetch recorded system logs	<b>getRecordedEdgeDeviceLogs</b>	✓
<b>GET</b>	/edgeDevices/{id}/systemVersion	Get System Version of EdgeDevice	<b>getSystemVersionOfEdgeDevice</b>	✓

## Customers



<b>POST</b>	/customers	Post a single new customer object	<b>createCustomer</b>	✓
-------------	------------	-----------------------------------	-----------------------	---

<b>GET</b>	<code>/customers</code>	Fetch all existing customers	<code>getAllCustomers</code>	▼
<b>GET</b>	<code>/customers/{id}</code>	Fetch single customer with given id	<code>getCustomer</code>	▼
<b>PUT</b>	<code>/customers/{id}</code>	Update a Customer (providing all fields)	<code>putCustomer</code>	▼
<b>PATCH</b>	<code>/customers/{id}</code>	Update a Customer (providing a subset of fields)	<code>patchCustomer</code>	▼
<b>DELETE</b>	<code>/customers/{id}</code>	Delete single customer with given id	<code>deleteCustomer</code>	▼
<b>GET</b>	<code>/customers/search/findByForeignId</code>	Find Customers by Foreign ID	<code>findCustomersByForeignId</code>	▼
<b>GET</b>	<code>/customers/{id}/machines</code>	Fetch all machines of a single customer with given id	<code>getMachinesOfCustomer</code>	▼
<b>POST</b>	<code>/customers/{id}/machines</code>	Create a new Machine for a single Customer	<code>createMachineForCustomer</code>	▼
<b>GET</b>	<code>/customers/{id}/users</code>	Get Users for a single Customer	<code>getUsersForCustomer</code>	▼
<b>POST</b>	<code>/customers/{id}/users</code>	Create User for a single Customer	<code>createUserForCustomer</code>	▼
<b>GET</b>	<code>/customers/{id}/sites</code>	Fetch all sites of a single customer with given id	<code>getSitesOfCustomer</code>	▼
<b>POST</b>	<code>/customers/{id}/sites</code>	Create a new site for a single customer with given id	<code>post_customers__id__sites</code>	▼
<b>POST</b>	<code>/customers/{id}/edgeDevices</code>	Create an edge device for a single customer with given id	<code>post_customers__id__edgeDevices</code>	▼
<b>GET</b>	<code>/customers/{id}/edgeDevices</code>	Fetch all edge devices of a single customer with given id	<code>getEdgeDevicesOfCustomer</code>	▼

## Machines



<b>GET</b>	<code>/machines</code>	Fetch all existing machines	<code>getAllMachines</code>	▼
<b>POST</b>	<code>/machines</code>	Create a new Machine	<code>createMachine</code>	▼
<b>POST</b>	<code>/machines/batch</code>	Create a batch of multiple machines	<code>createMachines</code>	▼

<b>DELETE</b>	/machines/batch	Delete a batch of multiple Machines	deleteMachines	▼
<b>GET</b>	/machines/{id}	Fetch single machine with given id	getMachine	▼
<b>PUT</b>	/machines/{id}	Update Machine	putMachine	▼
<b>PATCH</b>	/machines/{id}	Update Machine	patchMachine	▼
<b>DELETE</b>	/machines/{id}	Delete single machine with given id	deleteMachine	▼
<b>GET</b>	/machines/search/findByForeignId	Find Machines by Foreign ID	findMachinesByForeignId	▼
<b>GET</b>	/machines/{id}/customer	Fetch customer of single machine with given id	getCustomerOfMachine	▼
<b>GET</b>	/machines/{id}/edgeDevice	Fetch attached edge device of single machine with given id	getEdgeDeviceOfMachine	▼
<b>GET</b>	/machines/{id}/site	Fetch site of single machine with given id	getSiteOfMachine	▼

## Configurations



<b>GET</b>	/configurations	Fetch all existing configurations	getAllConfigurations	▼
<b>POST</b>	/configurations	Create a new Configuration	createConfiguration	▼
<b>POST</b>	/configurations/batch	Create a batch of multiple Configurations	createConfigurations	▼
<b>DELETE</b>	/configurations/batch	Delete a batch of multiple Configurations	deleteConfigurations	▼
<b>GET</b>	/configurations/{id}	Fetch single configuration by id	getConfiguration	▼
<b>PUT</b>	/configurations/{id}	Update single configuration by id	putConfiguration	▼
<b>PATCH</b>	/configurations/{id}	Update single fields of configuration by id	patchConfiguration	▼
<b>DELETE</b>	/configurations/{id}	Delete configuration by id	deleteConfiguration	▼

<b>GET</b>	/configurations/{id}/application	Fetch application of single configuration by id	getApplicationOfConfiguration	▼
------------	----------------------------------	---	-------------------------------	---

## Configuration Files ^

<b>GET</b>	/configurationFiles	Fetch all existing configuration files	getAllConfigurationFiles	▼
------------	---------------------	--	--------------------------	---

<b>POST</b>	/configurationFiles	Create a new ConfigurationFile	createConfigurationFile	▼
-------------	---------------------	--------------------------------	-------------------------	---

<b>GET</b>	/configurationFiles/search/findBy	Find ConfigurationFiles	findConfigurationFiles	▼
------------	-----------------------------------	-------------------------	------------------------	---

<b>GET</b>	/configurationFiles/{id}	Get ConfigurationFile	getConfigurationFile	▼
------------	--------------------------	-----------------------	----------------------	---

<b>PUT</b>	/configurationFiles/{id}	Update an existing ConfigurationFile	putConfigurationFile	▼
------------	--------------------------	--------------------------------------	----------------------	---

<b>PATCH</b>	/configurationFiles/{id}	Modify an existing ConfigurationFile	patchConfigurationFile	▼
--------------	--------------------------	--------------------------------------	------------------------	---

<b>DELETE</b>	/configurationFiles/{id}	Delete ConfigurationFile by id	deleteConfigurationFile	▼
---------------	--------------------------	--------------------------------	-------------------------	---

<b>GET</b>	/configurationFiles/{id}/application	Get Application to which this ConfigurationFile applies.	getApplicationOfConfigurationFile	▼
------------	--------------------------------------	--	-----------------------------------	---

## Applications ^

<b>GET</b>	/applications	Fetch all existing applications	getAllApplications	▼
------------	---------------	---------------------------------	--------------------	---

<b>POST</b>	/applications	Create a new application	createApplication	▼
-------------	---------------	--------------------------	-------------------	---

<b>GET</b>	/applications/{id}	Fetch a single application	getApplication	▼
------------	--------------------	----------------------------	----------------	---

<b>PATCH</b>	/applications/{id}	Modify an existing Application	patchApplication	▼
--------------	--------------------	--------------------------------	------------------	---

<b>DELETE</b>	/applications/{id}	Delete a single application	deleteApplication	▼
---------------	--------------------	-----------------------------	-------------------	---

<b>GET</b>	/applications/{id}/configurations	Fetch all configurations of single application with given id	getConfigurationsOfApplication	▼
------------	-----------------------------------	--	--------------------------------	---

<b>POST</b>	/applications/{id}/configurations	Add new configuration to a single application with given id	createConfigurationForApplication	▼
-------------	-----------------------------------	---	-----------------------------------	---

<b>GET</b>	/applications/{id}/configurationFiles	Fetch all associated configuration files	getConfigurationFilesOfApplication
<b>POST</b>	/applications/{id}/configurationFiles	Add new configuration file to a single application with given id	createConfigurationFileForApplication
<b>GET</b>	/applications/{id}/deploymentPreview	Get a preview of the configuration which would be applied when deploying this Application to the EdgeDevice with the given edgeDeviceId.	getApplicationDeploymentPreview
<b>GET</b>	/applications/{id}/deploymentConfigurations	Get a list of all Configurations which would be considered when deploying this Application to the EdgeDevice with the given edgeDeviceId. Note that when an environment or template variable is defined in more than one Configuration, only the variable definition of the last of these Configurations is effectively used for deployment.	getApplicationDeploymentVariables
<b>GET</b>	/applications/{id}/deploymentFiles	Get a list of all ConfigurationFiles which would be considered when deploying this Application to the EdgeDevice with the given edgeDeviceId. Note that this list may contain multiple ConfigurationFiles with same path and name combination. In this case, only the last ConfigurationFile will effectively be used for the deployment.	getApplicationDeploymentFiles
<b>GET</b>	/applications/{id}/deployments	Get a list of all Deployments containing this Application	getDeploymentsOfApplication

## Deployments ^

<b>GET</b>	/deployments	Fetch all existing deployments	getDeployments
<b>POST</b>	/deployments	Create a new deployment	createDeployment
<b>GET</b>	/deployments/{id}	Fetch a single deployments	getDeployment
<b>PATCH</b>	/deployments/{id}	Modify an existing Deployment	patchDeployment
<b>DELETE</b>	/deployments/{id}	Delete a single deployment	deleteDeployment



<b>GET</b>	/deployments/{id}/applications	Fetch all applications of a single deployment	getApplicationsOfDeployment	▼
------------	--------------------------------	---	-----------------------------	---

<b>GET</b>	/deployments/{id}/edgeDevices	Fetch all edge devices that are set to given deployment	getEdgeDevicesOfDeployment	▼
------------	-------------------------------	---	----------------------------	---

## System Versions

<b>GET</b>	/systemVersions	Fetch all existing system versions	getSystemVersions	▼
------------	-----------------	------------------------------------	-------------------	---

<b>POST</b>	/systemVersions	Add a new system version that should be the latest version for Edge Device upgrades (if version number is greater than latest)	addSystemVersion	▼
-------------	-----------------	--	------------------	---

<b>GET</b>	/systemVersions/latest	Fetch latest existing system version	getLatestSystemVersion	▼
------------	------------------------	--------------------------------------	------------------------	---

<b>GET</b>	/systemVersions/{id}	Fetch a single system versions	getSystemVersion	▼
------------	----------------------	--------------------------------	------------------	---

<b>GET</b>	/systemVersions/{id}/edgeDevices	Get EdgeDevices with given SystemVersion	getEdgeDevicesWithSystemVersion	▼
------------	----------------------------------	--	---------------------------------	---

## Sites

<b>GET</b>	/sites	Get all Sites	getAllSites	▼
------------	--------	---------------	-------------	---

<b>POST</b>	/sites	Create a new Site	createSite	▼
-------------	--------	-------------------	------------	---

<b>GET</b>	/sites/{id}	Fetch a single site	getSite	▼
------------	-------------	---------------------	---------	---

<b>PUT</b>	/sites/{id}	Update Site	putSite	▼
------------	-------------	-------------	---------	---

<b>PATCH</b>	/sites/{id}	Update Site	patchSite	▼
--------------	-------------	-------------	-----------	---

<b>DELETE</b>	/sites/{id}	Delete a single Site	deleteSiteById	▼
---------------	-------------	----------------------	----------------	---

<b>GET</b>	/sites/{id}/customer	Fetch customer of a single site	getCustomerOfSite	▼
------------	----------------------	---------------------------------	-------------------	---

<b>GET</b>	/sites/{id}/machines	Fetch all machines of a single site	getMachinesOfSite	▼
------------	----------------------	-------------------------------------	-------------------	---

<b>POST</b>	/sites/{id}/machines	Create a new machines and assign it to a single site (and customer)	createMachineForSite	▼
-------------	----------------------	---	----------------------	---

## Users



GET	/users	Get all Users	getAllUsers	▼
POST	/users	Create a User	createUser	▼
GET	/users/{id}	Get a single User	getUserById	▼
PUT	/users/{id}	Update a User	putUser	▼
PATCH	/users/{id}	Update a User (providing a subset of fields)	patchUser	▼
DELETE	/users/{id}	Delete a single User	deleteUserById	▼
GET	/users/{id}/customer	Get a Customer associated to a single User	getCustomerOfUser	▼

## Licenses



GET	/licenses	Get all Licenses	getAllLicenses	▼
POST	/licenses	Create a License	createLicense	▼
GET	/licenses/{id}	Get a single License	getLicenseById	▼
PUT	/licenses/{id}	Update a License	putLicense	▼
PATCH	/licenses/{id}	Update a License (providing a subset of fields)	patchLicense	▼
DELETE	/licenses/{id}	Delete a single License	deleteLicenseById	▼
GET	/licenses/{id}/customer	Get a Customer associated with a single License	getCustomerOfLicense	▼

## Containers



GET	/containers/{id}	Fetch a single container	getContainer	▼
-----	------------------	--------------------------	--------------	---

<b>PUT</b>	/containers/{id}	Update all container fields (only allowed when debugMode is enabled on containing edge device)	<b>putContainer</b>	∨
<b>PATCH</b>	/containers/{id}	Update specific container fields (only allowed when debugMode is enabled on containing edge device)	<b>patchContainer</b>	∨
<b>POST</b>	/containers/{id}/start	Start a single container on its edge device (as docker container)	<b>startContainer</b>	∨
<b>POST</b>	/containers/{id}/restart	Restart a single container on its edge device (as docker container)	<b>restartContainer</b>	∨
<b>POST</b>	/containers/{id}/stop	Stop a single container on its edge device (as docker container)	<b>stopContainer</b>	∨
<b>GET</b>	/containers/{id}/logs	Fetch system logs of single container since given timestamp	<b>getContainerLogs</b>	∨
<b>POST</b>	/containers/{id}/reset	Stop a single container on its edge device, clear all configuration directories and restart docker container	<b>resetContainer</b>	∨
<b>GET</b>	/containers/{id}/customConfigurationFiles	Fetch mounted files of single container (based on Application - ConfigurationFile)	<b>getContainerCustomFiles</b>	∨
<b>PUT</b>	/containers/{id}/customConfigurationFiles	Overwrite mounted files of this container (based on Application - ConfigurationFiles)	<b>putContainerCustomFiles</b>	∨
<b>POST</b>	/containers/{id}/customConfigurationFiles	Add mounted files of this container (based on Application - ConfigurationFiles), return array of merged files	<b>postContainerCustomFiles</b>	∨
<b>DELETE</b>	/containers/{id}/customConfigurationFiles	Remove mounted files of this container (based on Application - ConfigurationFiles), return array of remaining files	<b>deleteContainerCustomFiles</b>	∨
<b>GET</b>	/containers/{id}/edgeDevice	Get EdgeDevice on which this Container is on.	<b>getContainerEdgeDevice</b>	∨
<b>GET</b>	/containers/{id}/application	Get Application on which this Container is based on.	<b>getContainerApplication</b>	∨

## Accounting ∧

<b>GET</b>	/accounting/devices	Find productive EdgeDevices with attached production Machines for accounting	<b>findProductiveEdgeDevices</b>	∨
------------	---------------------	--	----------------------------------	---