

BVMS - Server and Configuration API

Author:

Wrobel Maciej (BT-VS/XSW-SEC)

Date:

11 January 2024

Table of Contents

1 Document information	3
1.1 Introduction	3
1.2 Capabilities and operations	3
1.3 Authentication	3
1.4 Authorization	3
1.5 Virtual Inputs API	4
1.6 Configuration API	5

1 Document information

Project	BVMS
Reference	n/a
Version	9
Last modified	2024, Jan 11 16:16

1.1 Introduction

The BVMS Server API provides a REST-based API for selected functionalities of the BVMS server as well as for some configuration options of BVMS. It is an optional feature of BVMS.

In order to use the BVMS Server API, please make sure, that during the installation one of the available setup options (depending on the BVMS version) is enabled: **Config API** or **Server API**.

1.2 Capabilities and operations

- Get all virtual inputs in the BVMS configuration
- Switch a virtual input on or off; send virtual input data
- Get all cameras in the BVMS configuration
- Modify (rename) cameras in the BVMS configuration
- Activate the BVMS configuration

1.3 Authentication

In order to access functionality of the Rest API you need to authenticate your request. At the moment only Basic Authentication Scheme is supported.

Access to the Configuration part of the API is mutually exclusive with any other client application used to configure BVMS (Configuration Client, Configuration Wizard). No access is granted if any of such application is currently logged on to the target BVMS.

If a specific Config API functionality requires any specific configuration permission, it will be explained in the corresponding section.

1.3.1 Basic Authentication Scheme

Standard Basic Authentication Scheme is supported according to [The 'Basic' HTTP Authentication Scheme](#). BVMS user name and password must be used. Authentication data must be provided in the

`Authorization` header of a request to the Server API:

```
Authorization: Basic <Base64 encoded<JoeDow:secret>>
```

1.4 Authorization

- Access to the Server API is granted when the user account belongs to a user group that has the application permission `BVMS Server SDK`.
- Access to the data of devices in BVMS is granted according to the `Device permissions` of the corresponding user group.

1.5 Virtual Inputs API

The overview of available commands for the Virtual Inputs is provided below.

GET /virtualinputs

Provides a paginated JSON array of all virtual inputs in the BVMS configuration.

Response includes all virtual inputs from the system for which the user has a permission (through the `Device permissions/Logical tree` of the corresponding user group in the Configuration Client).

GET /virtualinputs/{id}

Returns virtual input by its ID.

Request is fulfilled when the user has the permission for the virtual input defined in `Device permissions/Logical tree` for the corresponding user group. Otherwise the request will fail with '404 NotFound'.

GET /virtualinputs/{inputNumber}

Returns virtual input by its input number.

Request is fulfilled when the user has the permission for the virtual input defined in `Device permissions/Logical tree` for corresponding user group. Otherwise the request will fail with '404 NotFound'.

POST /virtualinputs/{id}/state

Switches on or switches off the specified virtual input, generating an according state event for this virtual input.

Request is fulfilled when the user has the permission for the virtual input defined in `Device permissions/Logical tree` for the corresponding user group.

POST /virtualinputs/{id}/textData

Generates a `Virtual Input Data Event` for the specified virtual input.

Request is fulfilled when the user has the permission for the virtual input defined in `Device permissions/Logical tree` for the corresponding user group.

POST /virtualinputs/{inputNumber}/state

Switches on or switches off the specified virtual input, generating an according state event for this virtual input.

Request is fulfilled when the user has the permission for the virtual input defined in `Device permissions/Logical tree` for the corresponding user group.

POST /virtualinputs/{inputNumber}/textData

Generates a `Virtual Input Data Event` for the specified virtual input.

Request is fulfilled when the user has the permission for the virtual input defined in `Device permissions/Logical tree` for the corresponding user group.

Further details, available parameters and examples are available in the YAML file, provided together with the BVMS. It can be found under:

- `/Bosch/VMS/Doc/APIs/ServerAPI` folder → for the Server API (Virtual Inputs)
- `/Bosch/VMS/Doc/APIs/ConfigAPI` folder → for the Configuration API

1.6 Configuration API

The overview of available commands for the BVMS Configuration is provided below.

GET /cameras

Provides a paginated JSON array of all cameras in the BVMS configuration.

Response includes all cameras from the system for which the user has a permission (through the `Camera permissions` of the corresponding user group / Enterprise access account in the Configuration Client).

PATCH /cameras

Modifies the list of all cameras of the BVMS configuration for which the caller has a permission, or only a subset thereof. Supported modifications include:

- Change camera display name

The Client MUST provide a set of modifications to apply to individual cameras in a JSON Patch document format. The Config API applies the entire set of changes atomically: either the entire JSON Patch document is applied successfully or nothing is changed. The JSON Patch document can not be applied when:

- It defines an operation to modify a camera that does not exist.
- It defines an operation to modify a camera for which the user lacks device permissions.
- A new camera display name is an illegal name (it is too long, it is empty or contains invalid characters)

- The number of patch operations exceeds the maximum number of operations in a single PATCH request.

POST /ConfigActivation

Activates the *latest* version of the BVMS configuration.

The BVMS configuration can be activated when:

- The latest configuration differs from the currently active configuration.
- There is no pending delayed activation already scheduled.

The client SHOULD NOT activate the BVMS configuration after each individual modification of the BVMS configuration (for example renaming of a single camera), but only perform a single activation for an entire set of related modifications.

Further details, available parameters and examples are available in the YAML file, provided together with the BVMS. It can be found under:

- */Bosch/VMS/Doc/APIs/ServerAPI* folder → for the Server API (Virtual Inputs)
- */Bosch/VMS/Doc/APIs/ConfigAPI* folder → for the Configuration API