

Gate ~ FMCU ~ management dashboard/en

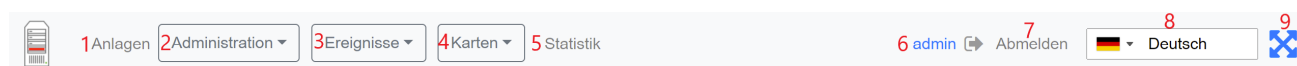


| Inhaltsverzeichnis | |
|----------------------------------|----|
| 1 General | 1 |
| 2 Headline | 1 |
| 3 Connectivity Board | 2 |
| 4 Configuration Management | 2 |
| 4.1 Architecture | 8 |
| 4.2 Roadmap | 9 |
| 5 Management of Events | 10 |
| 6 Events | 10 |
| 7 Event Definitions Page | 10 |
| 8 Event Definition Groups | 11 |
| 9 Event Subscriptions | 11 |
| 10 Configuration | 12 |

General

The Wanzl Access Manager offers a central dashbaord for access controls in a homogeneous interface, displays the corresponding states and supports the user in the administration of these components. The Wanzl Access Manager is a leading platform for monitoring and managing access controls of security areas.

Headline



| Nummer | Funktion |
|--------|------------------------------------------------------|
| 1 | Button Management Dashboard |
| 2 | Administration |
| 3 | Events |
| 4 | Cards |
| 5 | Statistics Button |
| 6 | Display from registered Users |
| 7 | Button to log out the User |
| 8 | Menu for selecting the language |
| 9 | Adjust the display to the entire width of the screen |

Connectivity Board

| Contact | Type | Impulse | Description |
|------------|--------|--------------|---------------------------------------|
| A8 | Output | Yes (500 ms) | Person has entered |
| A9 | Output | Yes (500 ms) | Alarms* |
| E3 | Input | Yes (500 ms) | Open Entry |
| E4 | Input | Yes (500 ms) | Open Exit |
| E6 | Input | Yes (200ms) | Card is valid, open the gate |
| E7 | Input | Yes (200ms) | Card is invalid, do not open the gate |
| E8 | Input | No | Emergency State button** |
| E9 | Input | Yes (500 ms) | Open Entry 70% |
| E10 | Input | No | Fire Alarm |

*A9 Alarms:

- Gate is out of order, when a mandatory device is broken or a connection to the server is broken (if the server is used in the solution).
- Tailgating or other unwanted actions.
- Entry barcode scanner is broken or it is a stub.
- Gate in in the **Fire Alarm** or **Emergency** state. The alarm is produces as a repetitive pulse signal (700ms).
- Gate is open only for exit.

After the Emergency State button pressed, the gate switches to the Emergency State. Then only exit from this state is to restart the gate.

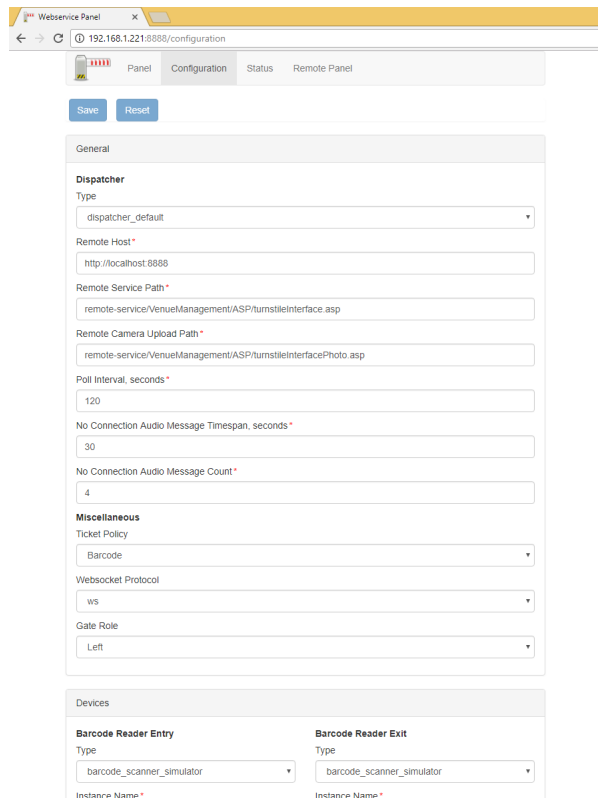
The signal can be set/received to/from a contact from any of the halves of the gate.

Configuration Management

User Interface

The Galaxy Gate is configured through the **/configuration** page. If the server is running at 192.168.1.221 and port is 8888, the url is:

<http://192.168.1.221/configuration>



There are two buttons on the page:

| Name | Description | Action |
|--------------|---------------------------------------------------------------------|--------|
| Save | Saves the modified configuration to the server | Post |
| Reset | Discards any changes by reloading the configuration from the server | Get |

The configuration is split into sections and subsections:

- General
 - Dispatcher
 - Miscellaneous
- Devices
 - Barcode Reader Entry
 - Barcode Reader Exit
 - Gate
 - Tracking Camera
 - Audio Player
 - Camera
- Displays
 - Display Gate Entry
 - Display Gate Exit

- Display Entry
- Display Exit
- Display TimeInfo

Here are some screenshots:

General

Dispatcher

Type
dispatcher_odoo_superfly ▼

Host*
192.168.1.221

Port*
8069

Protocol
http ▼

Password*
pass

Poll Interval, seconds*
120

No Connection Audio Message Timespan, seconds*
300

No Connection Audio Message Count*
4

Miscellaneous

Ticket Policy
Barcode ▼

Websocket Protocol
ws ▼

Gate Role
Left ▼

Devices

| | |
|----------------------------------------------------------------------------|---------------------------------------------------------------------|
| Barcode Reader Entry | Barcode Reader Exit |
| Type <input type="text" value="barcode_scanner_rfid"/> | Type <input type="text" value="barcode_scanner_magellan"/> |
| Instance Name* <input type="text" value="Barcode Scanner Entry"/> | Instance Name* <input type="text" value="Barcode Scanner Exit"/> |
| Read card not often than, ms* <input type="text" value="1000"/> | Read card not often than, ms* <input type="text" value="1000"/> |
| Port* <input type="text" value="/dev/ttyUSB2"/> | Port* <input type="text" value="/dev/ttyUSB3"/> |
| Beep Enabled <input type="checkbox"/> OFF | |
| Gate | |
| Type <input type="text" value="gate_galaxy_modbus"/> | |
| Port* <input type="text" value="/dev/ttyUSB0"/> | |
| Timeout, ms* <input type="text" value="500"/> | |
| Baud rate* <input type="text" value="115200"/> | |
| Reverse <input type="checkbox"/> OFF | |
| Timeout Count* <input type="text" value="2"/> | |
| Timespan Before Audio Message, seconds* <input type="text" value="10"/> | |

Audio Player

Type

audio_player_sonos

Host Ip Address *

192.168.101.86

Sonos Ip Addresses, separated by a comma *

192.168.101.159

Volume, % *

100

Camera

Type

camera_http

Uri *

192.168.1.33

User *

admin

Password *

pass

Delay On Enter, ms *

0

Delay On Exit, ms *

0

Disabled

OFF

Tracking Camera

Type

tracking_camera_xovis

Host*

10.122.21.21

Port*

80

Protocol

http

Password*

pass

Entry Line*

EnterLine

Exit Line*

ExitLine

Inner Entry Zone*

EnterDangerZone

Inner Exit Zone*

ExitDangerZone

Outer Entry Zone*

EnterZone

Outer Exit Zone*

ExitZone

Displays

| | |
|---------------------------------------------------------------|--------------------------------------------------------------|
| <input checked="" type="checkbox"/> Display Gate Entry | <input checked="" type="checkbox"/> Display Gate Exit |
| Content Module <input type="text" value="trade_fair"/> | Content Module <input type="text" value="default"/> |
| Rotate 90 Grad <input type="checkbox"/> OFF | Rotate 90 Grad <input type="checkbox"/> OFF |
| <input checked="" type="checkbox"/> Display Entry | <input checked="" type="checkbox"/> Display Exit |
| Show Video <input type="checkbox"/> OFF | Show Video <input type="checkbox"/> OFF |
| <input checked="" type="checkbox"/> Display Time Info | |

Facility Management Control Unit (v2.4-88-gf5bd36c-dirty)

A subsection is often has a structure with a combo box named **Type** at the top and some fields that follow. This structure is used for configuring an entity that can have several implementations. What is important is that the implementation can differ in type and number of controls. For example, a simulator of the gate has only two properties:

Gate

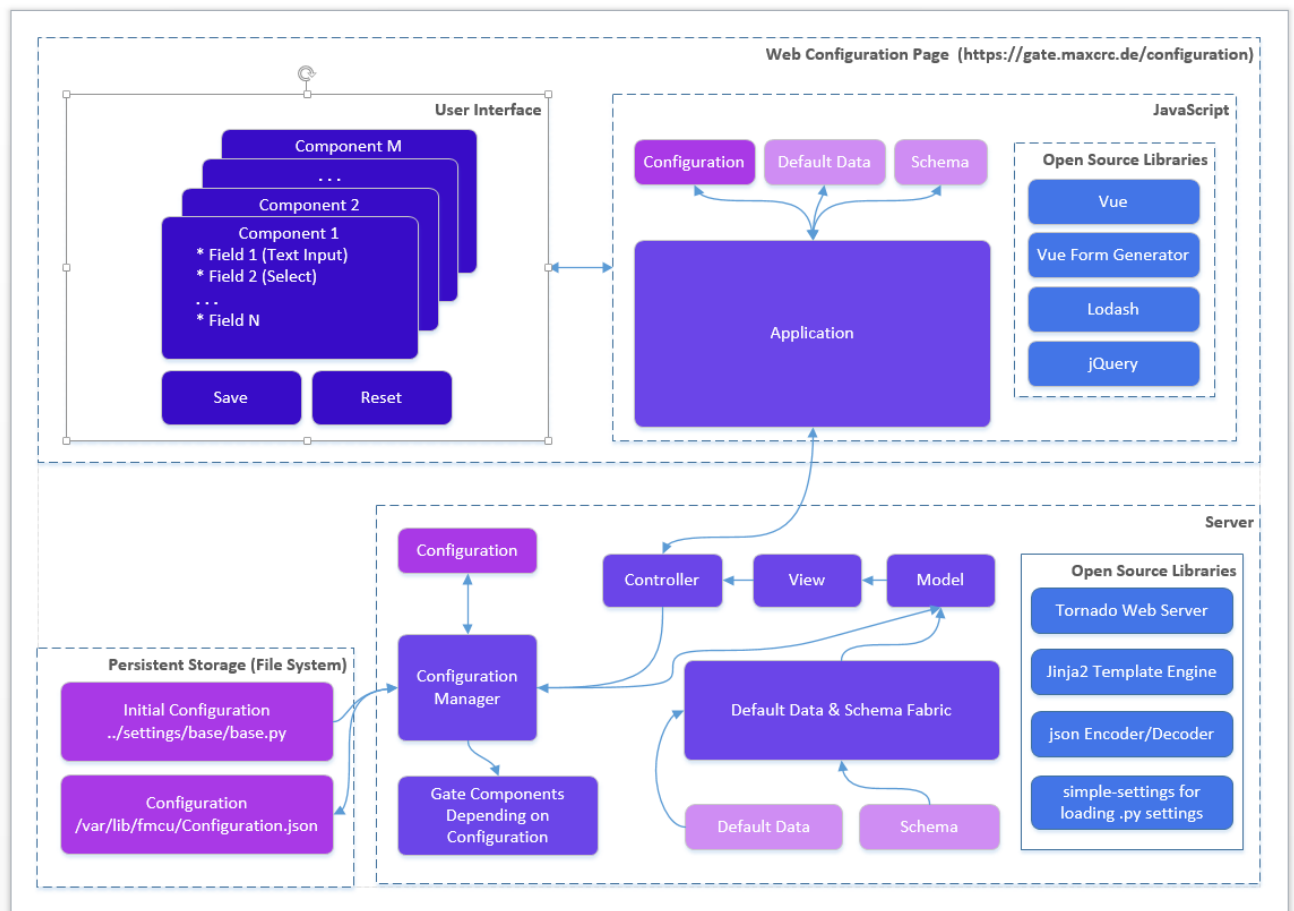
Type

Timeout Count*

Timespan Before Audio Message, seconds*

Architecture

The picture below gives a bird's eye view of the configuration management.



When the gate starts and the Configuration.json is present it is created from the Initial Configuration (base.py or other .py configuration file). The Schema contains a set of data for describing controls in form components:

- Which controls should contain a component
- Validations rules if any

E.g. a form component can contain a text box and an input for integers. The data range for integer should 1-100.

The Default Data contains the initial data for controls in form components.

Edit this section

Roadmap

The features are planned for the upcoming release:

- Configurations page allows you to see the list of available configurations. You can:
 - Clone an existing configuration to a new one
 - Delete a configuration
 - Make a configuration active. Only one configuration can be used by the server and it is called *active*.

- Apply button. This should apply changes on-the-fly.
- Restart button. This should restart the docker container and apply settings, which cannot be applied on-the-fly.
- Extend the vue form generator library:
 - Make the switch control consistent with the switch controls on other pages
 - Add a control which allows you to manage an array of items (e.g. the list of connected gates). E.g. <https://codepen.io/safx/pen/AhCtk>
- Import/Export of a Configuration.json.
- Reset any configuration to the default one (by removing the json file)

Management of Events

The following sections contains the term **Path**, which means the path is part of the URL. E.g., path **/events** for the FMCU Server running under **https://server.maxcrc.de/** means

```
https://server.maxcrc.de/events
```

Events

Web path:

```
/events
```

The page shows a filterable and sortable list of events:

| Name | Description |
|----------|----------------------------------------------|
| Date | Date and time when the event occurred |
| Name | Translated name of the event |
| Gate | Gate if appropriate where the event occurred |
| State | The state of the event if supported |
| Category | Category to which the event belongs to |
| Info | Additional info about the event |

Edit this section

Event Definitions Page

Path:

```
/event-definitions
```

The page shows a list of definitions of events:

| Name | Description |
|-------------------------|----------------------------------------------------------------------------------|
| Name | Name |
| Event Definition Groups | Groups to which the event belongs to if any |
| Supported States | Can be empty (for impulse events) or On, Off (for the events with states) |
| Category | Category to which the event belongs to |

Edit this section

Event Definition Groups

Path:

/event-definition-groups

The page shows a list of event groups:

| Name | Description |
|--------|-----------------------------------------|
| Name | Name |
| Events | Event definitions assigned to the group |
| Delete | Delete button |

An event definition group allows you to group one or more events so that it can be bound to a user group. See **Event Subscriptions** below

Edit this section

Event Subscriptions

Web path:

/event-subscriptions

The page shows a list of event definition groups bound to a user groups.

| Name | Description |
|------------------------|------------------------------------------------------------|
| Group | User group |
| Event definition group | Event definition group |
| Locked | When locked, events are not sent to the users of the group |
| Info | Any informative text |
| Delete | Delete button |

When a event definition group is bound to a user group and not locked, new events from the event definition group are sent to the users from the user group. The user must have a valid email address.

Example. A user group **Alarm events** with two members **user1** and **user2** is bound to an event definition group **Alarm events** with two events **GateTailgating** and **GateStateFireAlarm**. When either or both of these events occur, both users **user1** and **user2** receive an email notification about that.

Edit this section

Configuration

Path:

/configuration

In order the emails with events can be sent out, the **Email** section with the information like the hostname, port etc. of the outgoing server and credentials should be properly configured.