

# Covcheck

## Inhaltsverzeichnis

1 Introduction .....	1
2 Installation of Covcheck Application Server .....	1
2.1 Prerequisites .....	1
2.2 Installation Steps .....	1
2.3 Installation Files .....	3
3 Configuration of Covcheck Application Server .....	3
3.1 Change Default Password .....	3
3.2 Managing Rulesets .....	6
4 Configuration of Galaxy Gate .....	7
4.1 Dispatcher .....	7
4.2 Display .....	10

## Introduction

The document describes how to install the **Covcheck** application server and configure a Galaxy Gate so that COVID certificates and tests presented at the gate can be validated through the server's REST API.

## Installation of Covcheck Application Server

### Prerequisites

- Ubuntu 22.04 with Internet access
- User with **sudo** permissions and availability of a user with ID 1000 (**cat /etc/passwd | grep 1000**), which is by default
- Archive **covcheck-installer.tar.gz** with the files listed in [Installation Files](#). The name **covcheck-installer.tar.gz** can include a version, e.g. **covcheck-installer-v0.1.tar.gz**

### Installation Steps

- Connect with **ssh** to the Ubuntu server
- Copy **covcheck-installer.tar.gz** to the local directory
- Execute the following command:

```
tar xvf covcheck-installer.tar.gz && cd ./covcheck-installer
```

## Covcheck

- Modify file `.env` and specify the database password (**POSTGRES\_PASSWORD**) and, if necessary, host name (**ODOO\_HOST\_NAME**) and timezone (**TZ**):

```
POSTGRES_PASSWORD=gR7K6GT8MvQHpvLs
ODOO_HOST_NAME=covcheck.maxcr.de
TZ=Europe/Berlin
```

- If SSL certificates are provided for domain name **ODOO\_HOST\_NAME**, put them into directory **/etc/nginx/certs**, e.g:

```
sudo mkdir -p /etc/nginx/certs
sudo tar xvf certs.tar.gz -C /
```

- If SSL certificates are not available, HTTP connection will be used and you need to modify file `**docker-compose**` accordingly:

- Add this section to container `**odoo**`:

```
ports:
- 80:8069
```

- Comment out or remove port 80 for container `**nginx_proxy**`:

```
ports:
# - 80:80
- 443:443
```

- Run the installer script:

```
script=install-covcheck.sh && chmod +x $script && ./$script -i
```

- The installation is finished successfully if you see this line in the shell:

```
### Installation finished successfully
```

## Covcheck

You can make sure that the Covcheck is installed and running by executing the following command:

```
curl https://covcheck.maxcrc.de/covcheck/status
```

The reply must be a JSON like as follows:

```
{  
  "status": "ok",  
  "last_update": "2022-07-04 12:06:34"  
}
```

## Installation Files

---

### **.env**

environment variables

### **covcheck.tar.gz**

archived covcheck and request\_extension ODOO addons

### **docker-compose.yml**

docker compose file

### **git-submodule-status**

information about git submodules that comprise the Covcheck application server

### **install-covcheck.sh**

primary installation script

### **nginx-vhost**

template vhost file for nginx

### **odoo.conf**

configuration file for ODOO

## Configuration of Covcheck Application Server

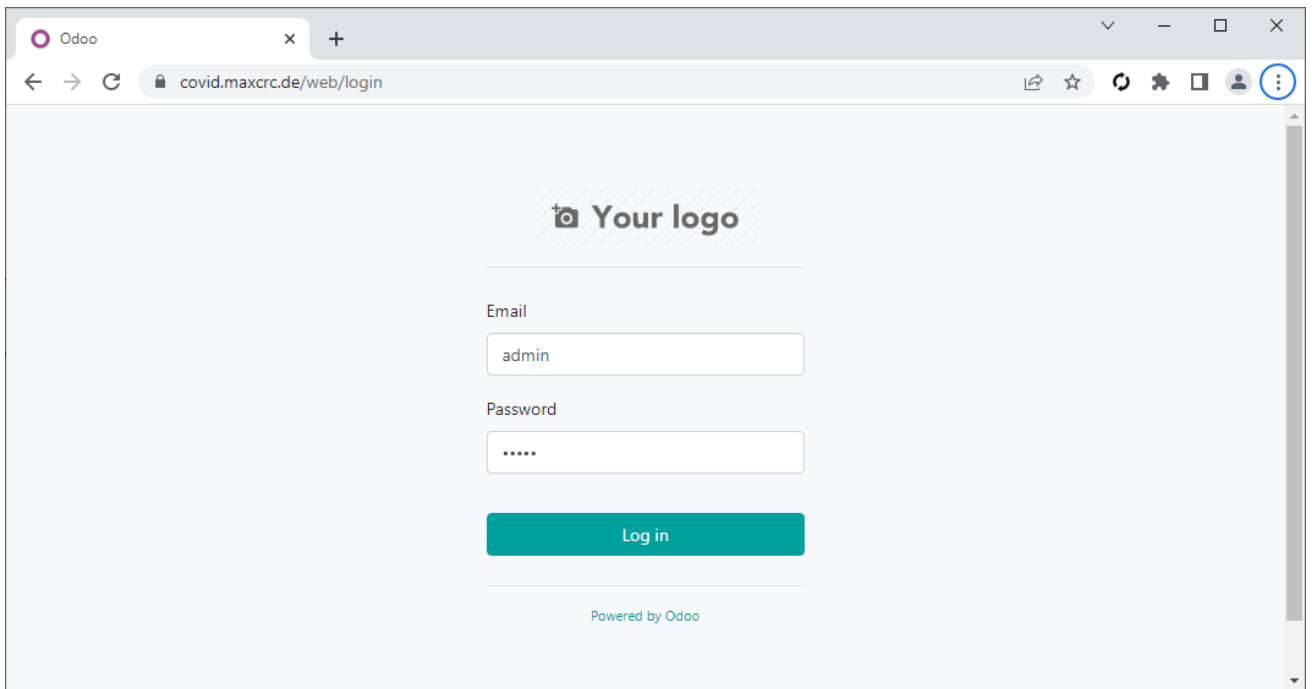
---

### Change Default Password

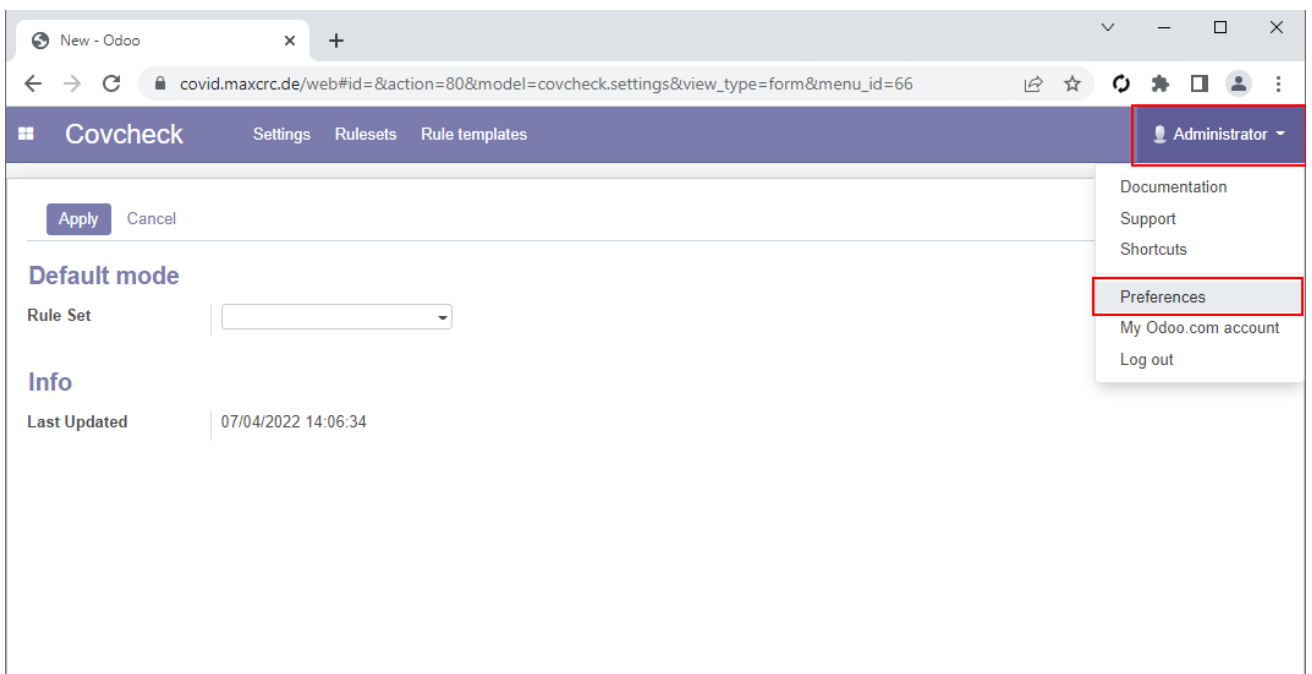
---

Open the address of the server in a browser and logging using the following default credentials:

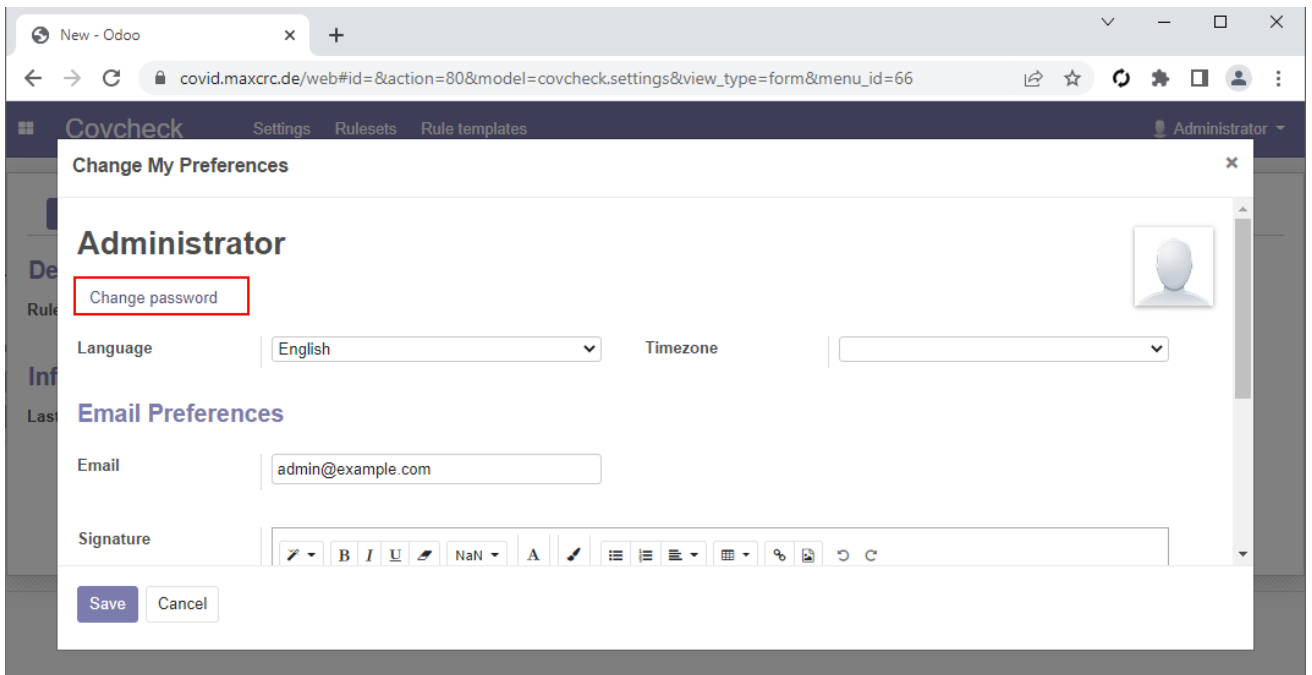
Username	Password
admin	admin



Click the **Administrator** in the top right corner and then **Preferences**:

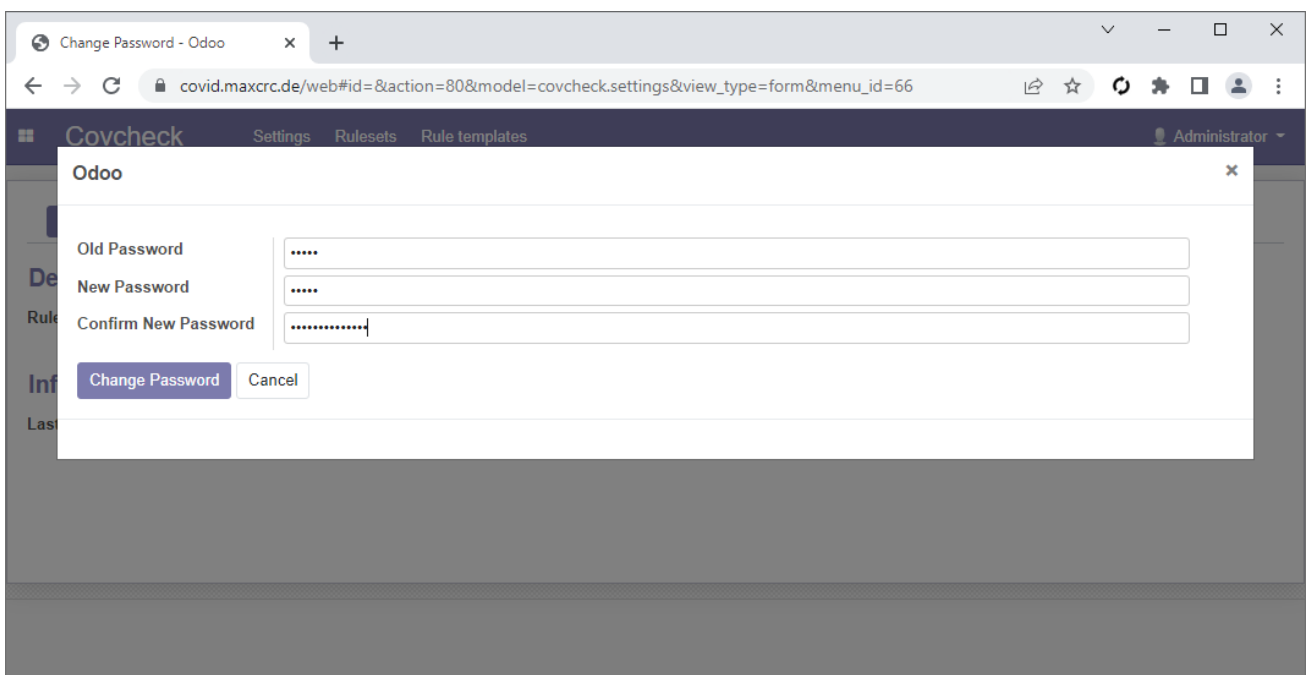


In the opened window click **Change password**:



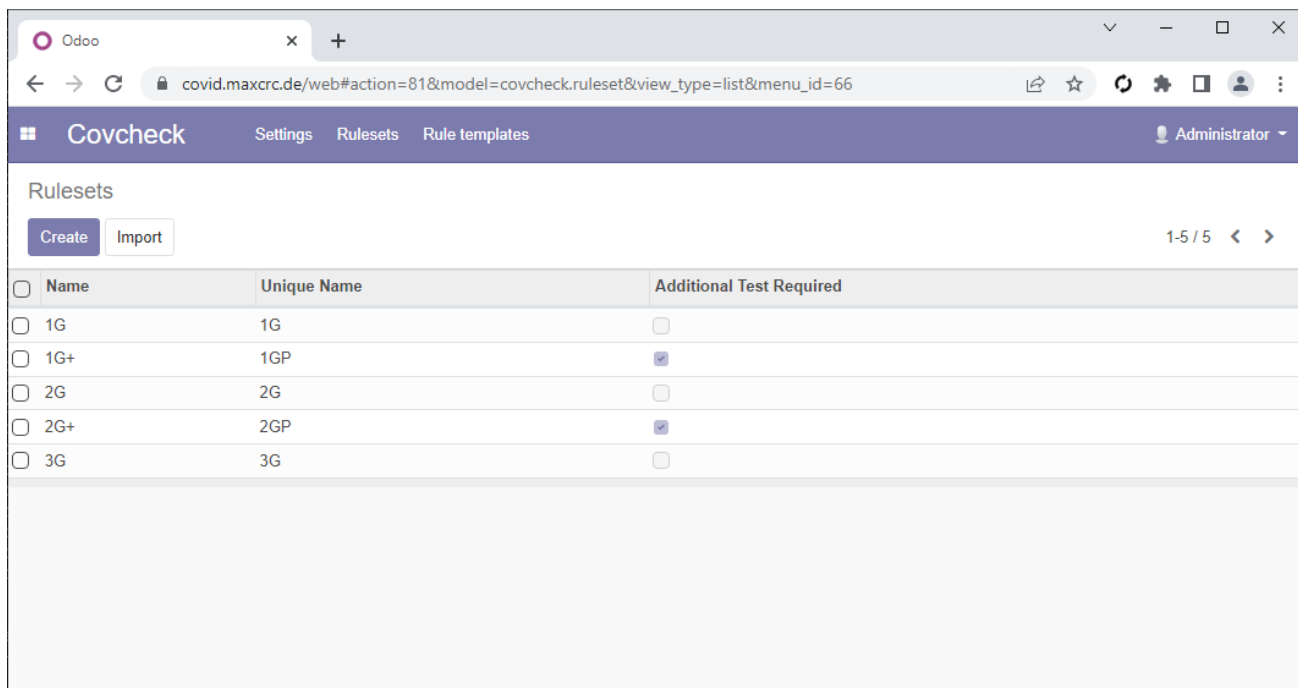
Fill-in the following fields and then click **Change Password**:

- Old Password
- New Password
- Confirm New Password



## Managing Rulesets

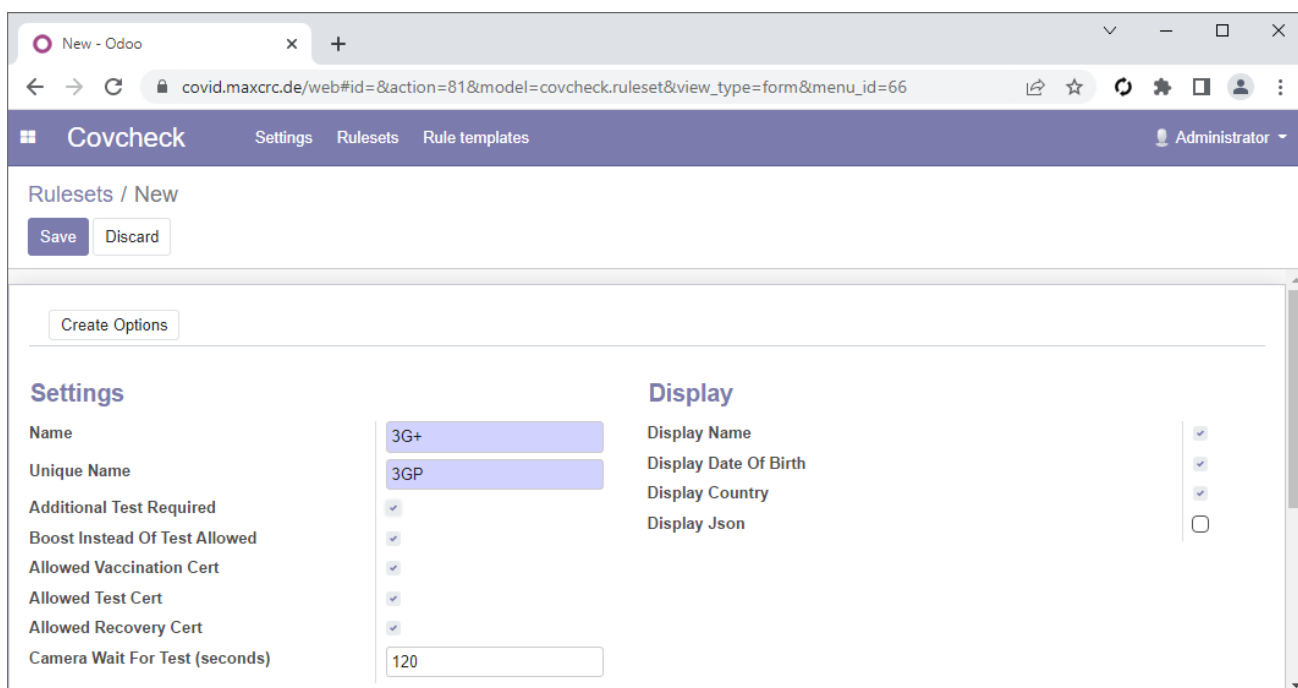
You can add, modify and delete rulesets on the **Rulesets** page:



The screenshot shows the Odoo web interface for the 'Covcheck' application. The browser address bar shows the URL: covid.maxcrc.de/web#action=81&model=covcheck.ruleset&view\_type=list&menu\_id=66. The application header includes 'Covcheck', 'Settings', 'Rulesets', and 'Rule templates', with the user 'Administrator' logged in. The main content area is titled 'Rulesets' and features 'Create' and 'Import' buttons. A table lists existing rulesets:

<input type="checkbox"/>	Name	Unique Name	Additional Test Required
<input type="checkbox"/>	1G	1G	<input type="checkbox"/>
<input type="checkbox"/>	1G+	1GP	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2G	2G	<input type="checkbox"/>
<input type="checkbox"/>	2G+	2GP	<input checked="" type="checkbox"/>
<input type="checkbox"/>	3G	3G	<input type="checkbox"/>

For example, let us create a ruleset **3G+**, which requires an additional test. Click **Create** fill-in the fields **Name** and **Unique Name**, check **Additional Test Required** and click **Save**:



The screenshot shows the 'Rulesets / New' form in the Odoo interface. The browser address bar shows the URL: covid.maxcrc.de/web#id=&action=81&model=covcheck.ruleset&view\_type=form&menu\_id=66. The application header is the same as in the previous screenshot. The form has 'Save' and 'Discard' buttons. A 'Create Options' button is visible. The form is divided into two sections: 'Settings' and 'Display'.

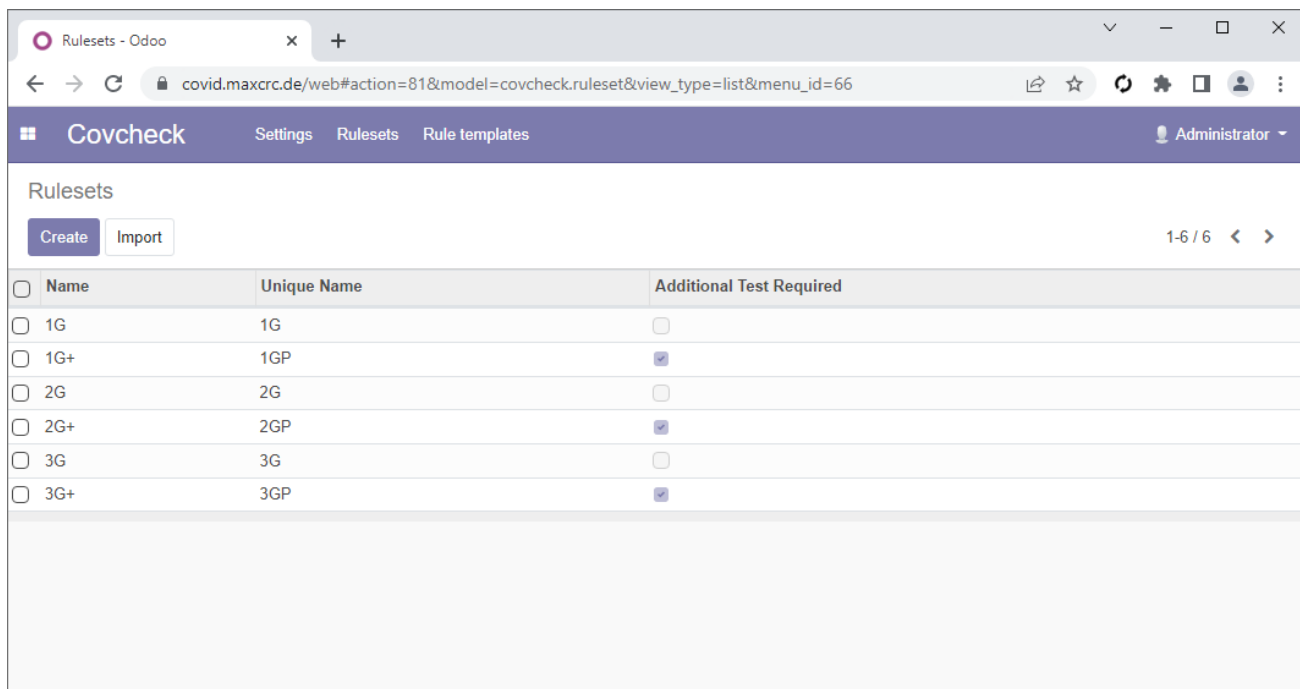
**Settings**

- Name: 3G+
- Unique Name: 3GP
- Additional Test Required:
- Boost Instead Of Test Allowed:
- Allowed Vaccination Cert:
- Allowed Test Cert:
- Allowed Recovery Cert:
- Camera Wait For Test (seconds): 120

**Display**

- Display Name:
- Display Date Of Birth:
- Display Country:
- Display Json:

The list now includes the **3G+** ruleset:



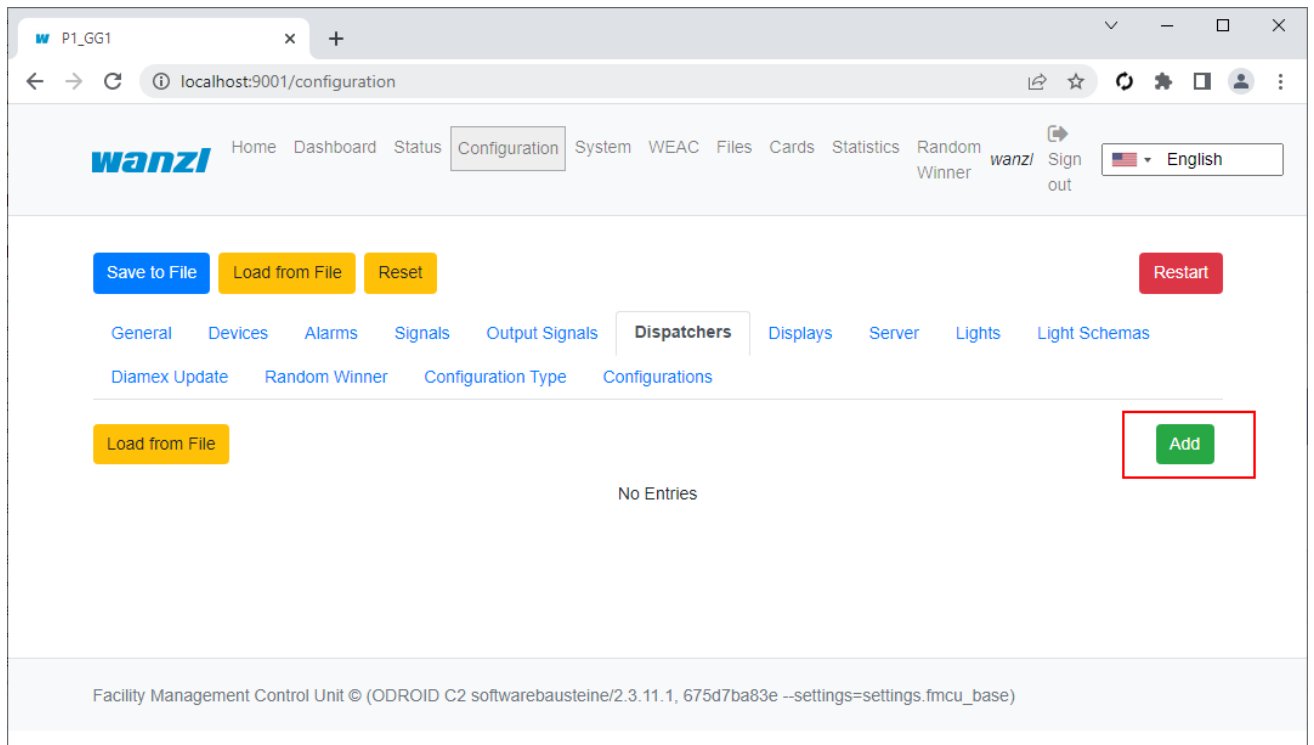
<input type="checkbox"/>	Name	Unique Name	Additional Test Required
<input type="checkbox"/>	1G	1G	<input type="checkbox"/>
<input type="checkbox"/>	1G+	1GP	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2G	2G	<input type="checkbox"/>
<input type="checkbox"/>	2G+	2GP	<input checked="" type="checkbox"/>
<input type="checkbox"/>	3G	3G	<input type="checkbox"/>
<input type="checkbox"/>	3G+	3GP	<input checked="" type="checkbox"/>

## Configuration of Galaxy Gate

In order COVID certificates and tests can be validated against the Covcheck server, the proper dispatcher and display must be selected and configured.

### Dispatcher

Open the URL of the gate, login with user **wanzl** and navigate to the **Configuration** page. Select tab **Dispatchers** and click **Add**:



In the combo box under the title **Dispatcher** select **Covcheck**, modify the following properties if needed and click **Save**:

**URL**

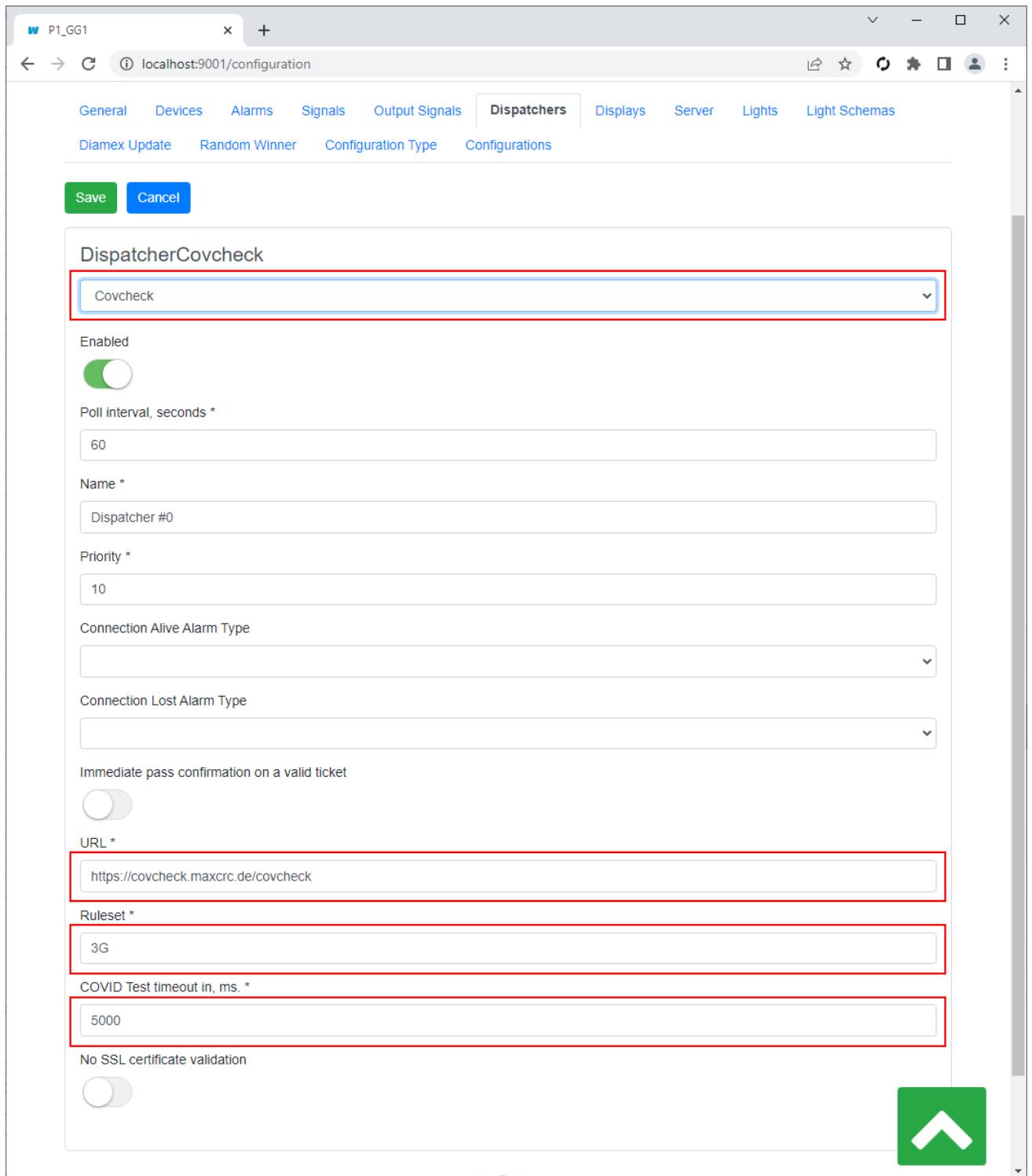
Base URL of Covcheck's API

**Ruleset**

Ruleset for validating COVID certificates and tests

**COVID Test timeout, ms**

When a COVID test is required, specifies the timeout of waiting for it



Browser window: P1\_GG1, localhost:9001/configuration

Navigation: General, Devices, Alarms, Signals, Output Signals, **Dispatchers**, Displays, Server, Lights, Light Schemas

Sub-navigation: Diamex Update, Random Winner, Configuration Type, Configurations

Buttons: Save, Cancel

DispatcherCovcheck

Covcheck

Enabled

Poll interval, seconds \*  
60

Name \*  
Dispatcher #0

Priority \*  
10

Connection Alive Alarm Type

Connection Lost Alarm Type

Immediate pass confirmation on a valid ticket

URL \*  
https://covcheck.maxcrc.de/covcheck

Ruleset \*  
3G

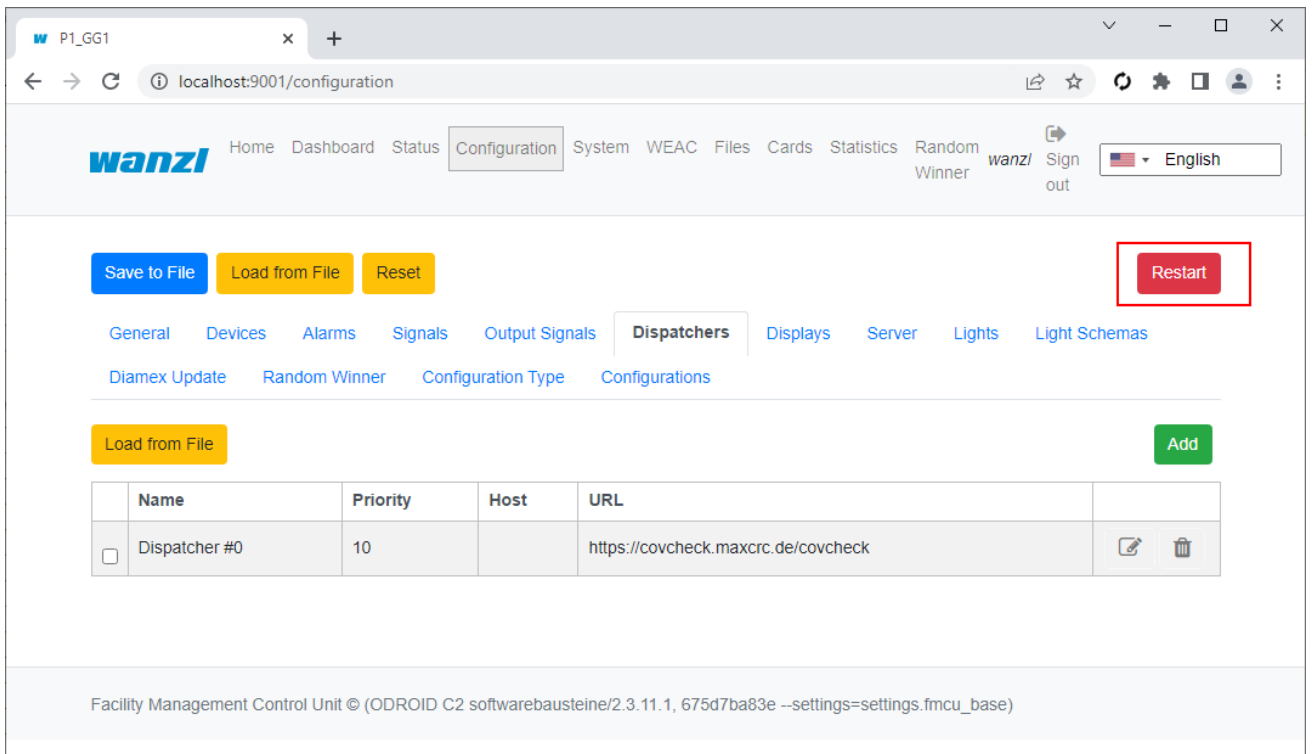
COVID Test timeout in, ms. \*  
5000

No SSL certificate validation

Green arrow button

Restart the gate:

## Covcheck





Save to File Load from File Reset Restart

General Devices Alarms Signals Output Signals Dispatchers Displays Server Lights Light Schemas

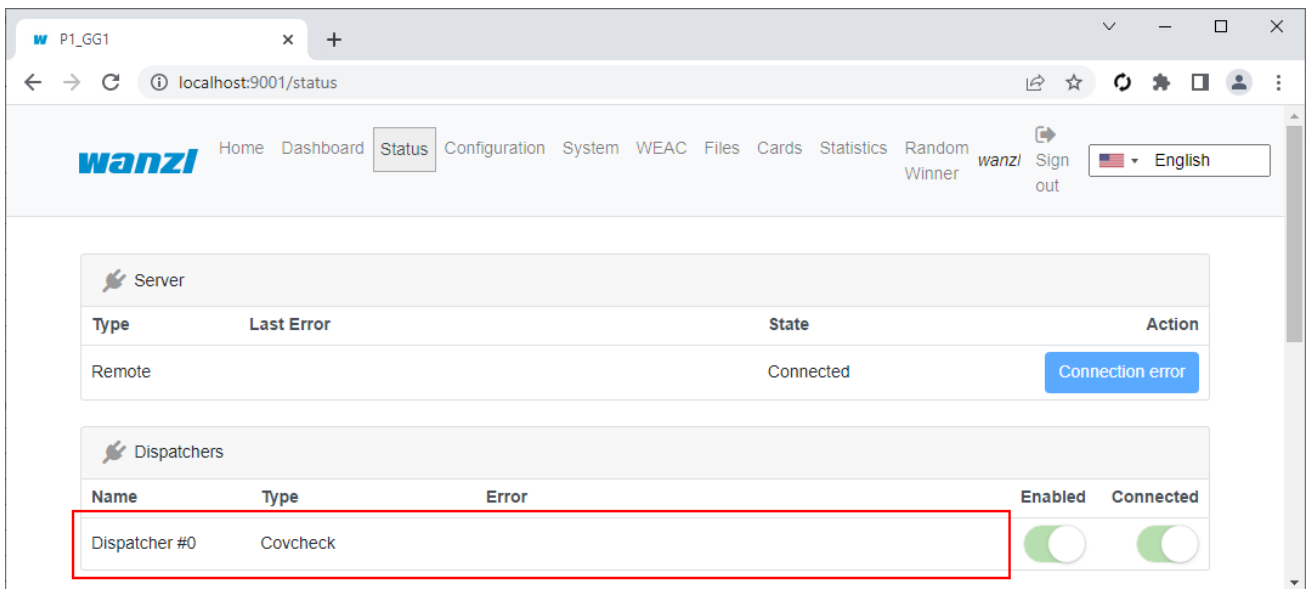
Diamex Update Random Winner Configuration Type Configurations

Load from File Add

	Name	Priority	Host	URL	
<input type="checkbox"/>	Dispatcher #0	10		https://covcheck.maxcrc.de/covcheck	 

Facility Management Control Unit © (ODROID C2 softwarebausteine/2.3.11.1, 675d7ba83e --settings=settings.fmcu\_base)

If the Covcheck server is available, the **Error** field is empty:



Server

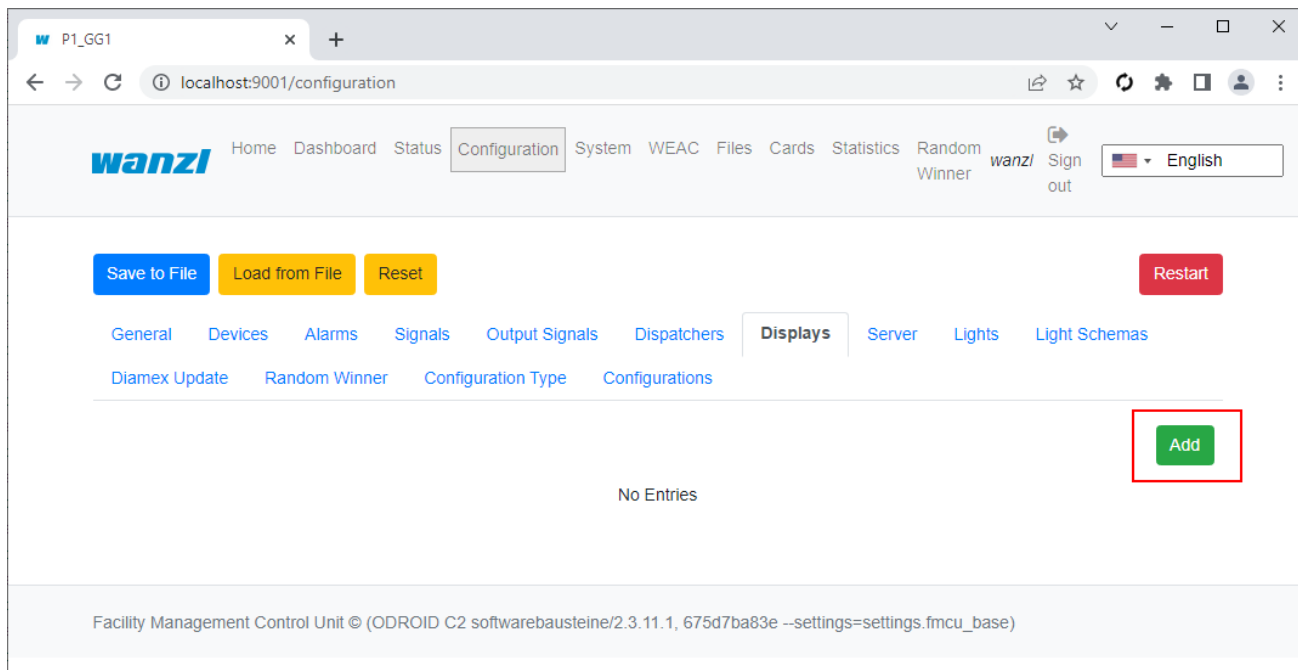
Type	Last Error	State	Action
Remote		Connected	Connection error

Dispatchers

Name	Type	Error	Enabled	Connected
Dispatcher #0	Covcheck		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

## Display

Open the URL of the gate, login with user **wanzl** and navigate to the **Configuration** page. Select tab **Displays** and click **Add**:



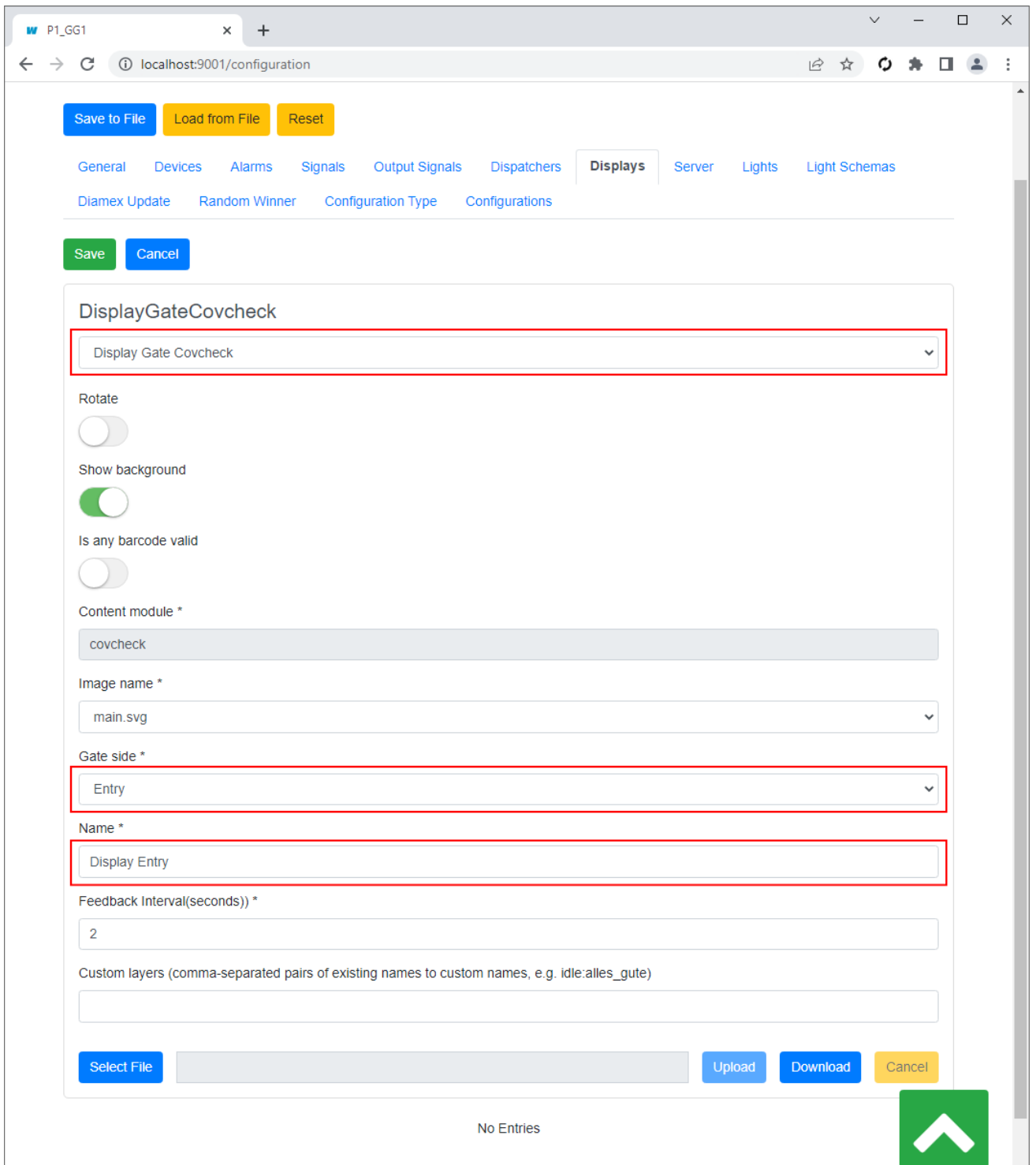
In the combo box under the title **Display** select **Display Gate Covcheck**, modify the following properties if needed and click **Save**:

**Gate side**

Side of the display, **Entry** or **Exit**

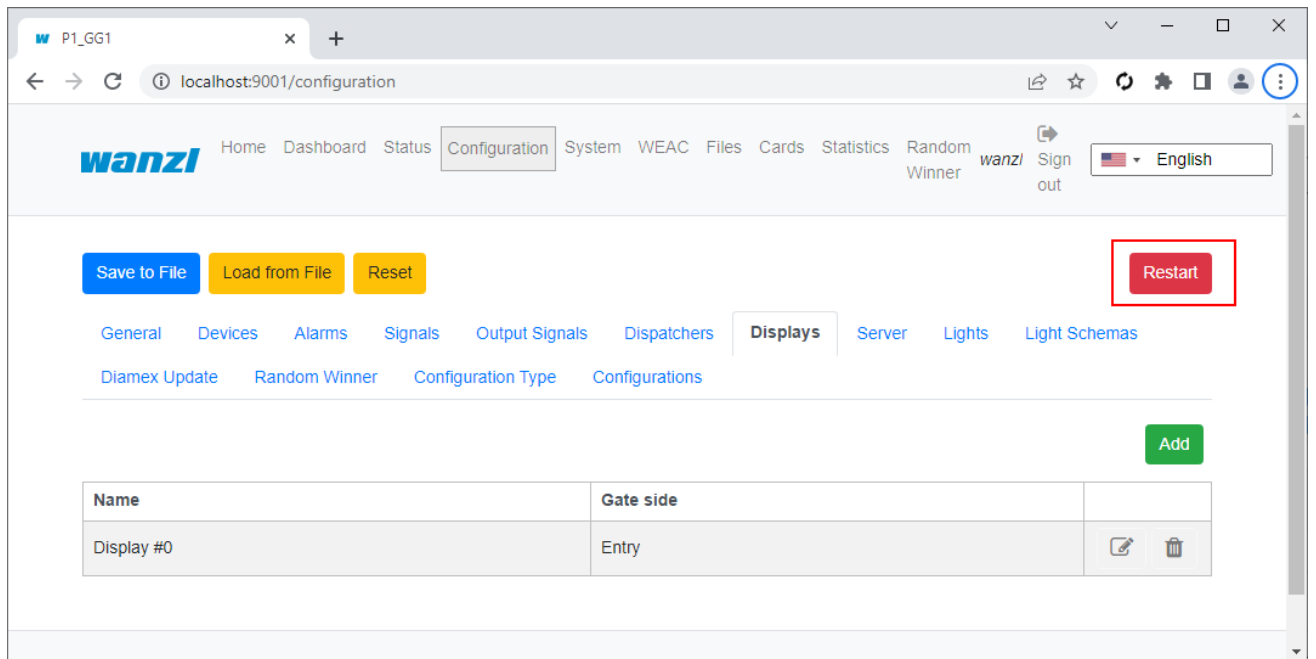
**Name**

User-friendly name

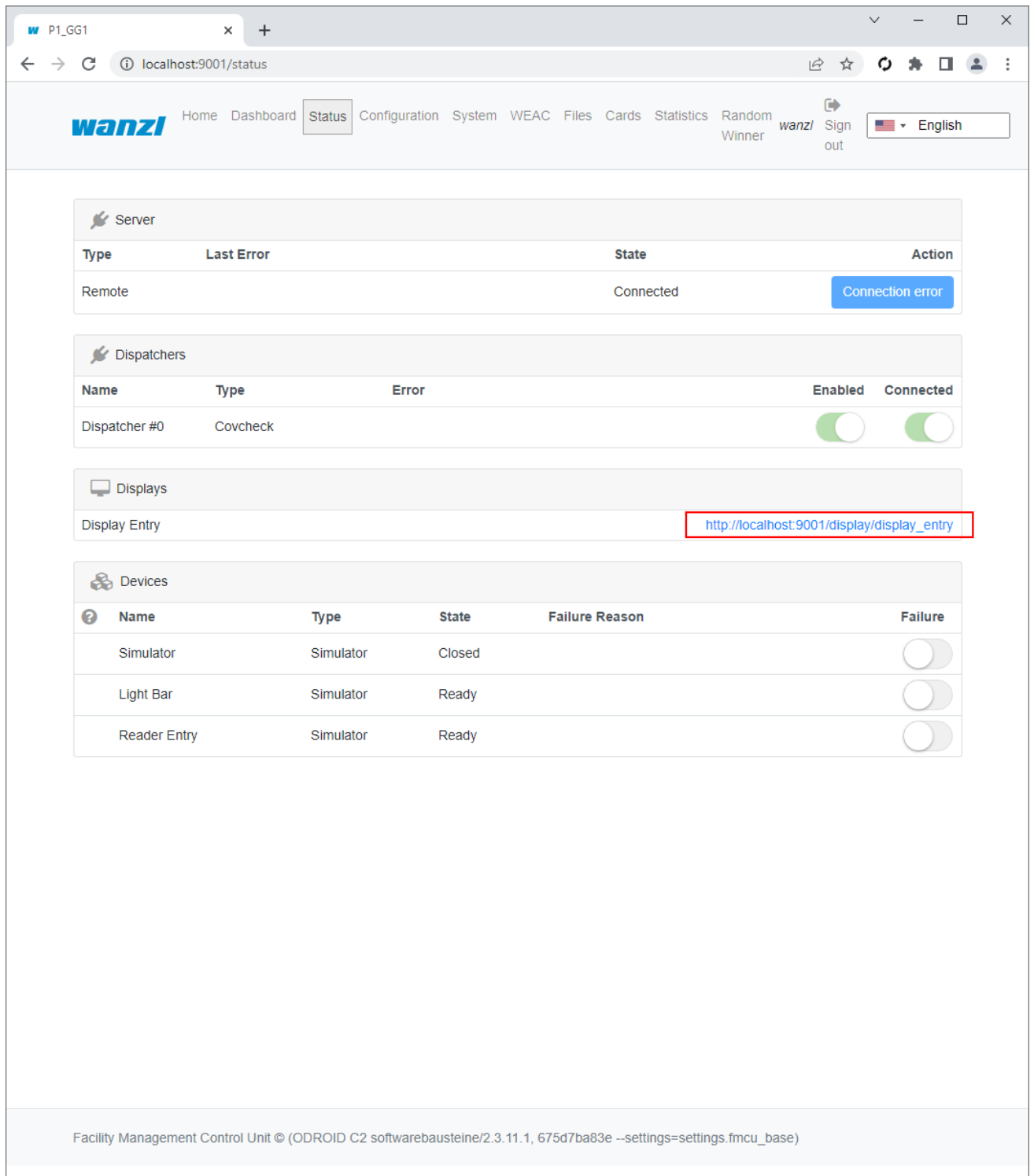


The screenshot shows a web browser window at localhost:9001/configuration. The interface includes navigation tabs for General, Devices, Alarms, Signals, Output Signals, Dispatchers, Displays (selected), Server, Lights, and Light Schemas. Below these are sub-tabs for Diamex Update, Random Winner, Configuration Type, and Configurations. At the top left, there are buttons for 'Save to File', 'Load from File', and 'Reset'. Below the navigation is a 'Save' button and a 'Cancel' button. The main configuration area is titled 'DisplayGateCovcheck' and contains several fields: a dropdown menu for 'Display Gate Covcheck' (highlighted with a red box), a 'Rotate' toggle (off), a 'Show background' toggle (on), an 'Is any barcode valid' toggle (off), a 'Content module \*' field with 'covcheck' selected, an 'Image name \*' dropdown with 'main.svg', a 'Gate side \*' dropdown with 'Entry' (highlighted with a red box), a 'Name \*' field with 'Display Entry' (highlighted with a red box), a 'Feedback Interval(seconds) \*' field with '2', and a 'Custom layers' field. At the bottom, there are buttons for 'Select File', 'Upload', 'Download', and 'Cancel'. A green arrow button is visible in the bottom right corner. The text 'No Entries' is displayed at the bottom center.

Restart the gate:



Navigate to the **Status** page and open the link corresponding to the display:



The screenshot shows a web browser window with the URL `localhost:9001/status`. The page is titled "wanzi" and has a navigation menu with items: Home, Dashboard, Status, Configuration, System, WEAC, Files, Cards, Statistics, Random Winner, and Sign out. The "Status" tab is active. The page content is organized into four main sections:

- Server:** A table with columns "Type", "Last Error", "State", and "Action". It shows one entry: "Remote" with state "Connected" and an "Action" button labeled "Connection error".
- Dispatchers:** A table with columns "Name", "Type", "Error", "Enabled", and "Connected". It shows one entry: "Dispatcher #0" with type "Covcheck", "Enabled" and "Connected" toggle switches both turned on.
- Displays:** A table with columns "Display Entry" and "URL". The "Display Entry" is "Display Entry" and the "URL" is `http://localhost:9001/display/display_entry`, which is highlighted with a red box.
- Devices:** A table with columns "Name", "Type", "State", "Failure Reason", and "Failure". It shows three entries: "Simulator" (State: Closed), "Light Bar" (State: Ready), and "Reader Entry" (State: Ready). Each entry has a "Failure" toggle switch.

At the bottom of the page, there is a footer: "Facility Management Control Unit © (ODROID C2 softwarebausteine/2.3.11.1, 675d7ba83e --settings=settings.fmcu\_base)".

The following screen for the idle state invites to present a COVID certificate:





When in addition a test certificate is required, the next screen asks to show it:



The screenshot shows a web browser window with the address bar containing "localhost:9001/display/display\_er" and "localhost:9001/display/display\_entry". The main content area has a light blue background and features a blue icon of a hand holding a smartphone. Below the icon, the text "Bitte Test-Zertificat scannen" is displayed in a large, bold, blue font. At the bottom, the number "5" is followed by an hourglass icon, indicating a 5-minute time limit.

