PyScada Documentation

Release 0.7.0rc16

Martin Schröder

Jun 25, 2019

Installation and Commandline

A Open Source SCADA System with HTML5 HMI, build using the Django framework. If you like to setup your own _SCADA_ system head over to the *Installation* section.

Important: This Version of PyScada is BETA software and may have serious bugs which may cause damage to your computer, automation hardware and data. It is not intended for use in production systems! You use this Software on your own risk!

CHAPTER 1

Installation

This installation guide covers the installation of PyScada for Debian 7/8/9 , Raspbian, Fedora 22/23 based Linux systems using MySQL / MariaDB or SQLite as Database, Gunicorn as WSGI HTTP Server and nginx as HTTP Server.

1.1 Dependencies

1.1.1 Debian 9, Raspbian

```
sudo -i
apt-get update
apt-get -y upgrade
# if you use MariaDB/MySQL as Database system (recommend)
apt-get -y install mariadb-server python3-mysqldb
apt-get install -y python3-pip libhdf5-100 libhdf5-dev python3-dev nginx

pip3 install gunicorn
pip3 install pyserial
pip3 install docutils
```

1.1.2 macOS

- *MySQL Server <https://www.mysql.de/>*
- HDF5 TODO

```
brew install python
export PATH=$PATH:/usr/local/mysql/bin
pip install MySQL-python
```

1.1.3 all

```
sudo -i
pip3 install https://github.com/trombastic/PyScada/archive/dev/0.7.x.zip

# for VISA Protocol
pip3 install pyvisa pyvisa-py
# for 1Wire Protocol
apt-get install owfs #
pip3 install pyownet
# for smbus Protocol, install libffi-dev first!
apt-get install libffi-dev
pip3 install smbus-cffi
```

1.2 Add a new system-user for Pyscada (optional, recommend)

Add a dedicated user for the pyscada server instance and add a directory for *static/media* files.

1.2.1 Linux

```
sudo -i
useradd -r pyscada
mkdir -p /var/www/pyscada/http
chown -R pyscada:pyscada /var/www/pyscada
mkdir -p /home/pyscada
chown -R pyscada:pyscada /home/pyscada
```

1.2.2 macOS

```
sudo -i
dscl . -create /Users/pyscada IsHidden 1
dscl . -create /Users/pyscada NFSHomeDirectory /Users/pyscada
LastID=`dscl . -list /Users UniqueID | awk '{print $2}' | sort -n | tail -1`
NextID=$((LastID + 1))
dscl . create /Users/pyscada UniqueID $NextID
dscl . create /Users/pyscada PrimaryGroupID 20
mkdir -p /var/www/pyscada/http
chown -R pyscada:staff /var/www/pyscada/
```

1.3 Create a MySql Database

Create the Database and grand the nessesery permission. Replace *PyScada_db*, *PyScada-user* and *PyScada-user-password* as you like.

```
mysql -uroot -p -e "CREATE DATABASE PyScada_db CHARACTER SET utf8;GRANT ALL␣
↪PRIVILEGES ON PyScada_db.* TO 'PyScada-user'@'localhost' IDENTIFIED BY 'PyScada-
↪user-password';"
```

1.4 Create a new Django Project

```
# Linux/OSX
cd /var/www/pyscada/
sudo -u pyscada django-admin startproject PyScadaServer
```

see *Django Settings* for all necessary adjustments to the django settings.py and urls.py.

1.5 Initialize Database And Copy Static Files

```
cd /var/www/pyscada/PyScadaServer # linux
sudo -u pyscada python3 manage.py migrate
sudo -u pyscada python3 manage.py collectstatic

# load fixtures with default configuration for chart lin colors and units
sudo -u pyscada python3 manage.py loaddata color
sudo -u pyscada python3 manage.py loaddata units

# initialize the background service system of pyscada
sudo -u pyscada python3 manage.py pyscada_daemon init
```

1.6 Add a Admin User To Your Django Project

```
cd /var/www/pyscada/PyScadaServer
sudo -u pyscada python3 manage.py createsuperuser
```

1.7 Setup the Webserver (nginx, gunicorn)

```
# debian
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/nginx_
↪sample.conf -O /etc/nginx/sites-available/pyscada.conf

# Fedora
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/nginx_
↪sample.conf -O /etc/nginx/conf.d/pyscada.conf
```

after editing, enable the configuration and restart nginx, optionally remove the default configuration

1.8 to use ssl (https, recommend)

generate ssl certificates.

```
# for Debian, Ubuntu, Raspian
sudo mkdir /etc/nginx/ssl
# the certificate will be valid for 5 Years,
sudo openssl req -x509 -nodes -days 1780 -newkey rsa:2048 -keyout /etc/nginx/ssl/
↪pyscada_server.key -out /etc/nginx/ssl/pyscada_server.crt
```

```
# debian
sudo ln -s /etc/nginx/sites-available/pyscada.conf /etc/nginx/sites-enabled/
sudo rm /etc/nginx/sites-enabled/default
```

now it's time to [re]start nginx.

```
# systemd (Debian 8, Fedora, Ubuntu > XX.XX)
sudo systemctl enable nginx.service # enable autostart on boot
sudo systemctl restart nginx

# SysV-Init (Debian 7, Ubuntu <= XX.XX, [Debian 8])
sudo service nginx restart
```

for Fedora you have to allow nginx to serve the static and media folder.

```
sudo chcon -Rt httpd_sys_content_t /var/www/pyscada/http/
```

add gunicorn and pyscada unit files:

```
# systemd
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/
↪service/systemd/gunicorn.socket -O /etc/systemd/system/gunicorn.socket
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/
↪service/systemd/gunicorn.service -O /etc/systemd/system/gunicorn.service
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/
↪service/systemd/pyscada_daemon.service -O /etc/systemd/system/pyscada.service

# in some installations gunicorn is not at /usr/local/bin/gunicorn but at /usr/bin/
↪gunicorn
# in this case you have to change the pat in the file /etc/systemd/system/gunicorn.
↪service accordingly

# enable the services for autostart
sudo systemctl enable gunicorn
sudo systemctl start gunicorn
sudo systemctl enable pyscada
```

1.9 Start PyScada

```
sudo systemctl start pyscada
```

CHAPTER 2

Django Settings

2.1 urls.py

Open the urls configuration file and add the necessary rewrite rule to the django URL dispatcher.

```
nano /var/www/pyscada/PyScadaServer/PyScadaServer/urls.py
```

```
...
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^', include('pyscada.hmi.urls')),
]
...
```

2.2 settings.py

Open the django settings file and make the following modifications. See also the django documentation for more Information.

```
nano /var/www/pyscada/PyScadaServer/PyScadaServer/settings.py
```

First deactivate the debugging, if debugging is active django will keep all SQL queries in the ram, the data-acquisition runs a massive amount of queries so your system will run fast out of memory. Keep in mind to restart gunicorn and the pysada daemons after you change the debugging state.

```
DEBUG = False
```

Add the host/domain of your machine, in this case every url that point to a ip of the machine is allowed.

```
ALLOWED_HOSTS = ['*']
```

Add PyScada and the PyScada sub-apps to the installed apps list of Django.

```
INSTALLED_APPS = [
    ...
    'pyscada.core',
    'pyscada.modbus',
    'pyscada.phant',
    'pyscada.visa',
    'pyscada.hmi',
    'pyscada.systemstat',
    'pyscada.export',
    'pyscada.onewire',
    'pyscada.smbus',
]
```

To use the MySQL Database, fill in the database, the user and password as selected in the *create Database section*.

```
DATABASES = {
    'default': {
        'ENGINE':   'django.db.backends.mysql',
        'NAME':     'PyScada_db',
        'USER':     'PyScada-user',
        'PASSWORD': 'PyScada-user-password',
        'OPTIONS': {
            'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",
        }
    }
}
```

Set the static file and media dir as follows.

```
...
STATIC_URL = '/static/'

STATIC_ROOT = '/var/www/pyscada/http/static/'

MEDIA_URL = '/media/'

MEDIA_ROOT = '/var/www/pyscada/http/media/'
```

Add all PyScada specific settings, keep in mind to set the file right file encoding in the *settings.py* file header (see also https://www.python.org/dev/peps/pep-0263/).

```
#!/usr/bin/python
# -*- coding: <encoding name> -*-
```

Append to the end of the *settings.py*:

```
# PyScada settings
# https://github.com/trombastic/PyScada

# email settings
DEFAULT_FROM_EMAIL = 'example@host.com'
EMAIL_HOST = 'mail.host.com'
EMAIL_PORT = 587
```

(continues on next page)

(continued from previous page)

```
EMAIL_HOST_USER = 'pyscada@host.com'
EMAIL_USE_TLS = True
EMAIL_USE_SSL = False
EMAIL_HOST_PASSWORD = 'password'
EMAIL_PREFIX = 'PREFIX' # Mail subject will be "PREFIX subjecttext"

# meta information's about the plant site
PYSCADA_META = {
    'name':'A SHORT NAME',
    'description':'A SHORT DESCRIPTION',
}

# export properties
#
PYSCADA_EXPORT = {
    'file_prefix':'PREFIX_',
    'output_folder':'~/measurement_data_dumps',
}

# View Options
#
LINK_TARGET = '_blank' # '_blank' for new tab or '_self' for opening it in the same␣
↪window

LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'formatters': {
        'standard': {
            'format' : "[%(asctime)s] %(levelname)s [%(name)s:%(lineno)s] %(message)s
↪",
            'datefmt' : "%d/%b/%Y %H:%M:%S"
        },
    },
    'handlers': {
        'file': {
            'level': 'DEBUG',
            'class': 'logging.FileHandler',
            'filename': BASE_DIR + '/pyscada_debug.log',
            'formatter': 'standard',
        },
    },
    'loggers': {
        'django': {
            'handlers': ['file'],
            'level': 'INFO',
            'propagate': True,
        },
        'pyscada': {
            'handlers': ['file'],
            'level': 'DEBUG',
            'propagate': True,
        },
    },
}
```

CHAPTER 3

Nginx Setup

3.1 nginx configuration

```
# debian
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/nginx_
↪sample.conf -O /etc/nginx/sites-available/pyscada.conf
sudo nano /etc/nginx/sites-available/pyscada.conf
# Fedora
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/nginx_
↪sample.conf -O /etc/nginx/conf.d/pyscada.conf
sudo nano /etc/nginx/conf.d/pyscada.conf
```

after editing, enable the configuration and restart nginx, optionaly remove the default configuration

```
# debian
sudo ln -s /etc/nginx/sites-available/pyscada.conf /etc/nginx/sites-enabled/
sudo rm /etc/nginx/sites-enabled/default
```

3.2 to use ssl

generate ssl certificates.

```
# for Debian, Ubuntu, Raspian
sudo mkdir /etc/nginx/ssl
# the certificate will be valid for 5 Years,
sudo openssl req -x509 -nodes -days 1780 -newkey rsa:2048 -keyout /etc/nginx/ssl/
↪pyscada_server.key -out /etc/nginx/ssl/pyscada_server.crt
```

now it's time to [re]start nginx.

```
# SysV-Init
sudo service nginx restart
# systemd
systemctl enable nginx.service # enable autostart on boot
sudo systemctl restart nginx
```

for Fedora you have to allow nginx to serve the static and media folder.

```
sudo chcon -Rt httpd_sys_content_t /var/www/pyscada/http/
```

Important: This Version of PyScada is BETA software and may have serious bugs which may cause damage to your computer, automation hardware and data. It is not intended for use in production systems! You use this Software on your own risk!

CHAPTER 4

Docker

This guide covers the basic setup of PyScada with Docker and Docker Compose.

4.1 Download the necessary files

First of all download the docker config files for building your images.

Using Git.

```
git clone https://github.com/trombastic/PyScada.git
cd PyScada/docker
```

Using wget.

```
wget -qO- -O tmp.zip https://github.com/trombastic/PyScada/archive/dev/0.7.x.zip &&␣
↪unzip tmp.zip && rm tmp.zip
cd PyScada-dev-0.7.x/docker
```

4.2 Generating SSL Certificates

Generate ssl certificates for using ssl.

```
cd nginx/ssl
openssl req -x509 -nodes -days 1780 -newkey rsa:2048 -keyout ./pyscada_server.key -
↪out ./pyscada_server.crt
```

4.3 Build and Run the Image

Build the PyScada Docker Image.

```
sudo docker-compose build
```

After the Images have been successfully build we need to initialize the Database and Create a superuser.

```
sudo docker-compose run pyscada /src/pyscada/pyscada_init
sudo docker-compose run pyscada /src/pyscada/manage.py createsuperuser
```

The last step is to start the Container.

```
sudo docker-compose up -d
```

CHAPTER 5

0.6.x to 0.7.x

Sorry a direct upgrade is not possible, you have to install 0.7.x from scratch.

CHAPTER 6

0.7.0b18 to 0.7.0b19

```
cd /var/www/pyscada/PyScadaServer
sudo -u pyscada python manage.py migrate
sudo -u pyscada python manage.py collectstatic
sudo -u pyscada python manage.py pyscada_daemon init
```

CHAPTER 7

systemd

```
sudo wget https://raw.githubusercontent.com/trombastic/PyScada/dev/0.7.x/extras/
↪service/systemd/pyscada_daemon.service -O /etc/systemd/system/pyscada_daemon.service
sudo systemctl enable pyscada_daemon
sudo systemctl disable pyscada_daq
sudo systemctl disable pyscada_event
sudo systemctl disable pyscada_mail
sudo systemctl disable pyscada_export
sudo rm /lib/systemd/system/pyscada_daq.service
sudo rm /lib/systemd/system/pyscada_mail.service
sudo rm /lib/systemd/system/pyscada_export.service
sudo rm /lib/systemd/system/pyscada_event.service
sudo systemctl daemon-reload
```

CHAPTER 8

Command-line

8.1 Start the PyScada Daemons

SysV-init and upstart:

```
service pyscada_daq_daemon start
service pyscada_daemon start
service gunicorn_pyscada start
```

systemd:

```
systemctl start "pyscada_*"
```

Django manage command:

```
python manage.py PyScadaDaemonHandler daemon_name start
```

8.2 Start Gunicorn

SysV-init and upstart:

```
service gunicorn_django start
```

systemd:

```
systemctl start gunicorn.service
```

8.3 Get Installed PyScada Version

```
cd ~/www/PyScadaServer
python manage.py shell
import pyscada
pyscada.__version__
exit()
```

8.4 Export Recorded Data Tables

```
python manage.py PyScadaExportData # last 24 houres
python manage.py PyScadaExportData "01-Mar-2015 00:00:00" # from 01. of March until␣
↪now
# from 01. of March until now, with the given filename
python manage.py PyScadaExportData "01-Mar-2015 00:00:00" "filename.h5"
# from 01. of March until 10. of March, with the given filename
python manage.py PyScadaExportData "01-Mar-2015 00:00:00" "filename.h5" "10-Mar-2015␣
↪00:00:00"
```

CHAPTER 9

Using the Front-End

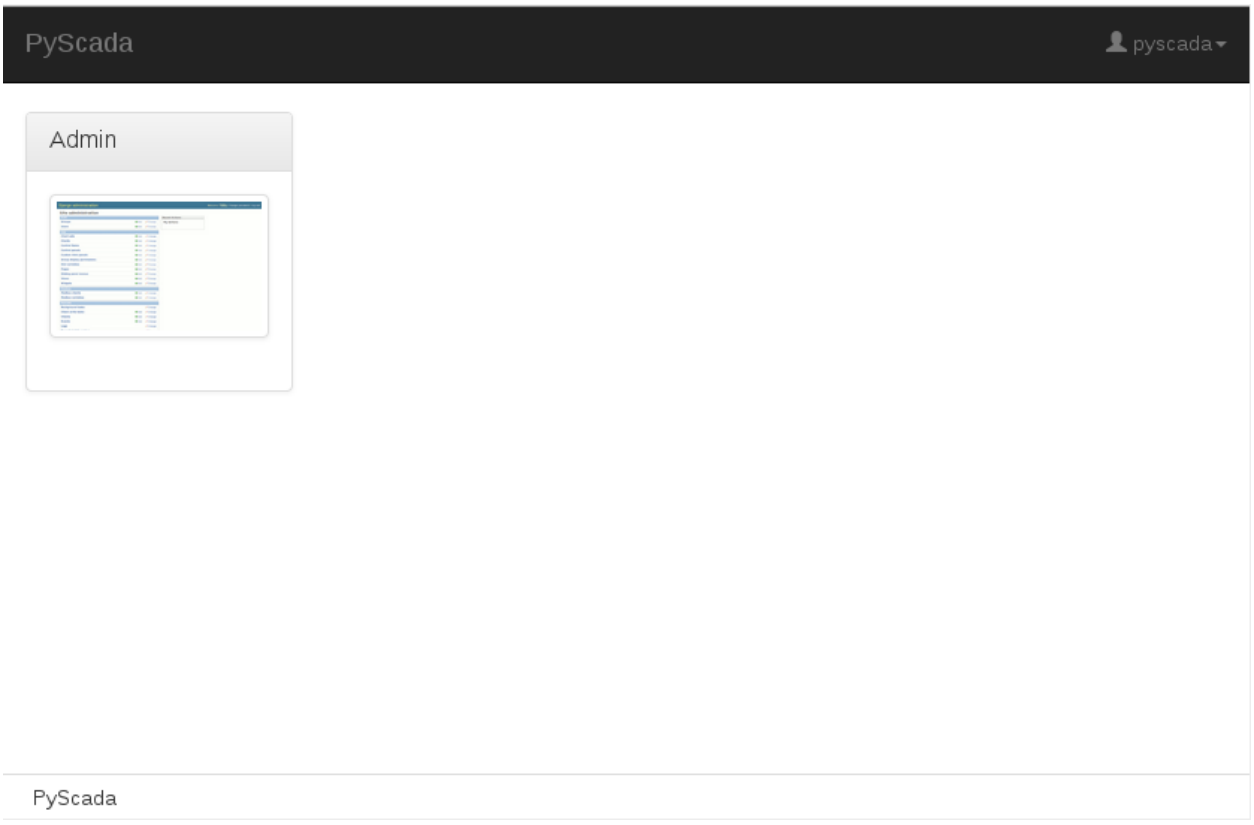
to be done…

CHAPTER 10

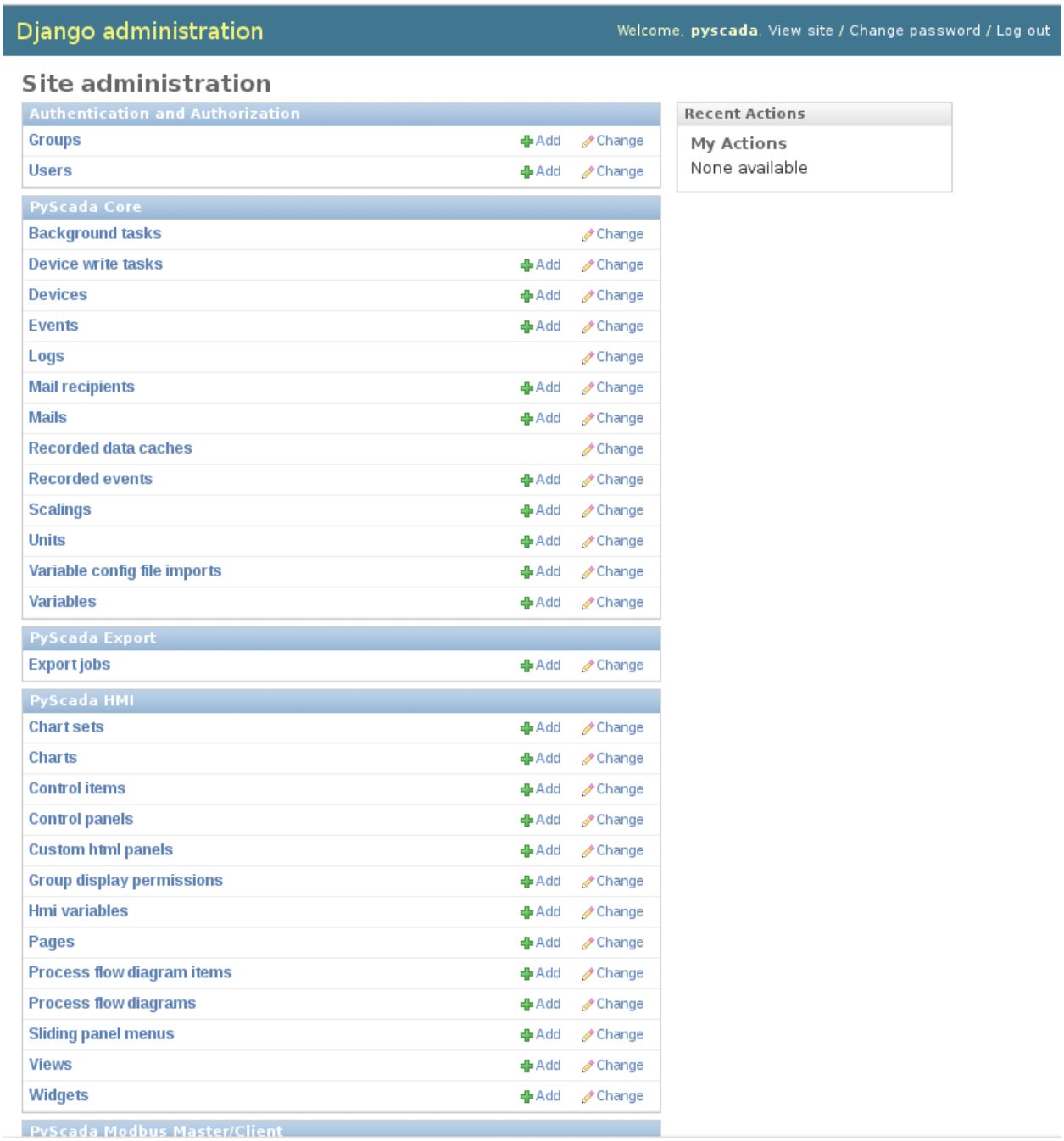
Using the Backend

To use the backend open the HMI in your browser by opening http://127.0.0.1 (replace 127.0.0.1 withe the IP or hostname of your PyScada server) and sign in with your admin account (*TODO link to createsuperuser* doc).

After successful login in your see the view overview, to open the admin panel click on the *Admin* Icon.



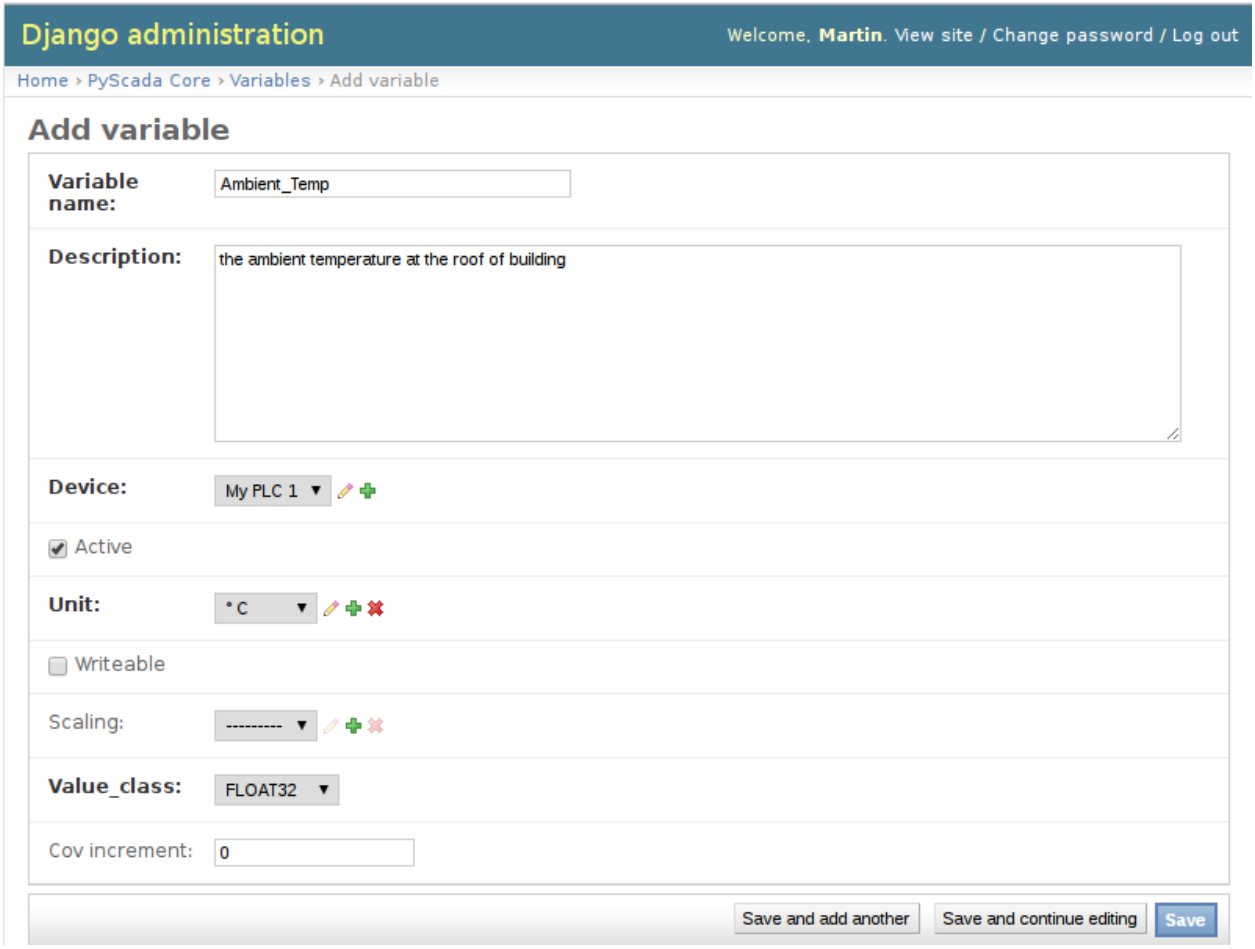
Now you are in the backend or Admin panel.



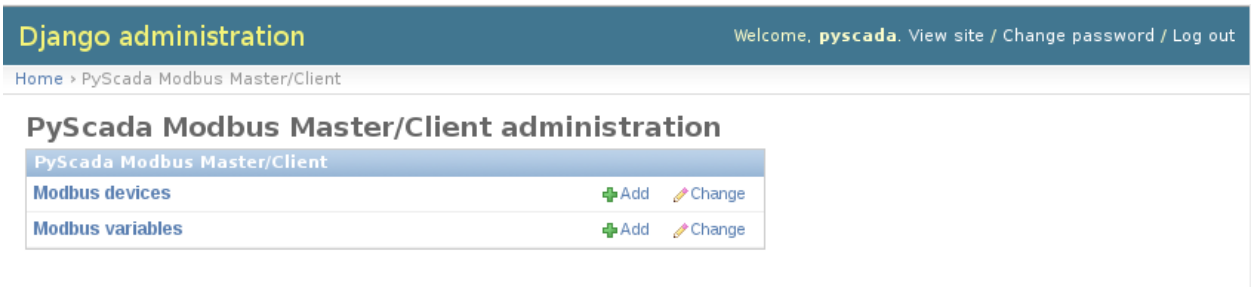
10.1 Add a new Device

To add a new device (e.g. a PLC) open the *Device* Table in the *PyScada Core* section.

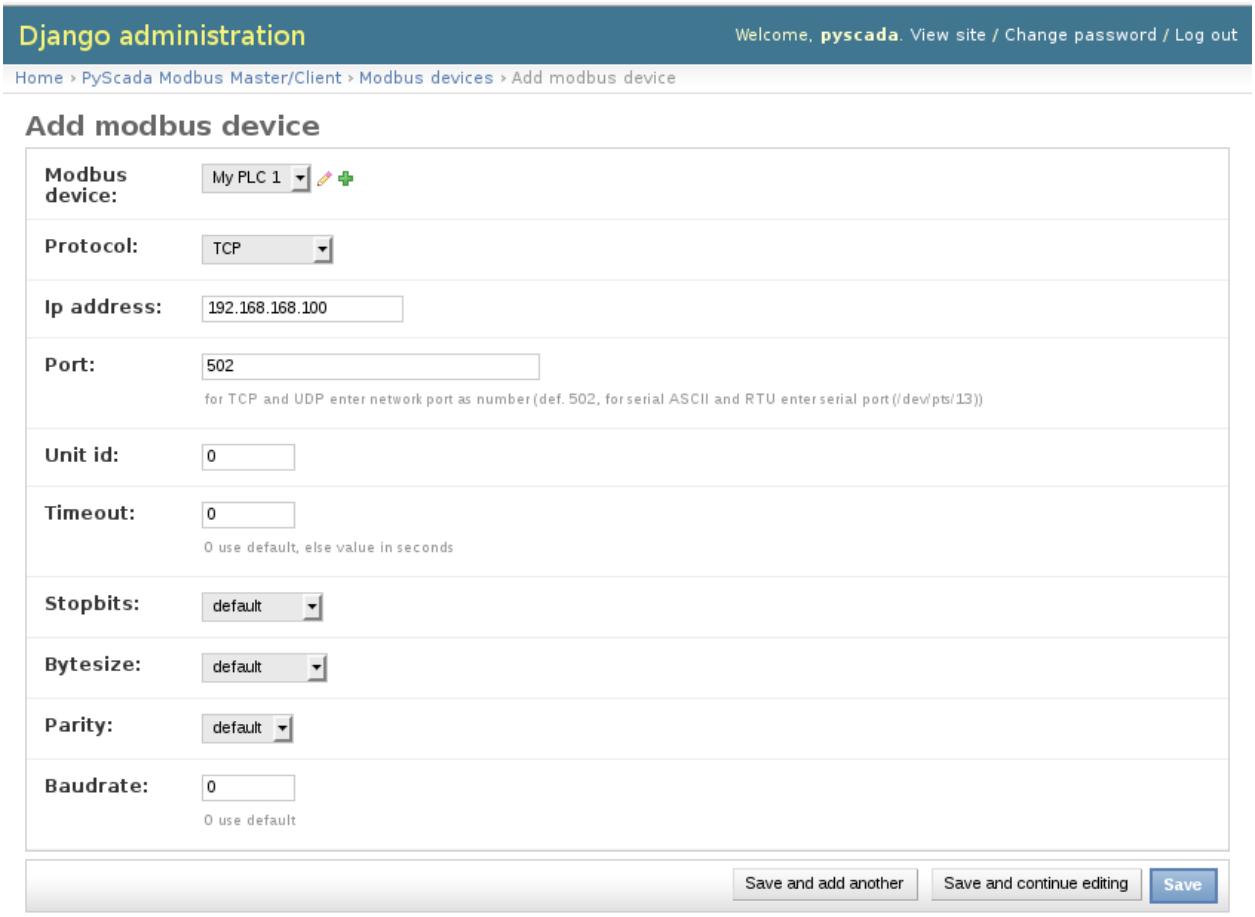
You will see a empty list, click on *add device* in the upper right corner to add a new device (e.g. a modbus device).



We assume we want to add a modbus TCP/IP device (the device act as modbus server, while pyscada act as client). The invormation related to the modbus device will be set in a seperate table. Go to the Admin overview and select *Modbus device* from the *PyScada Modbus Master/Client* section.



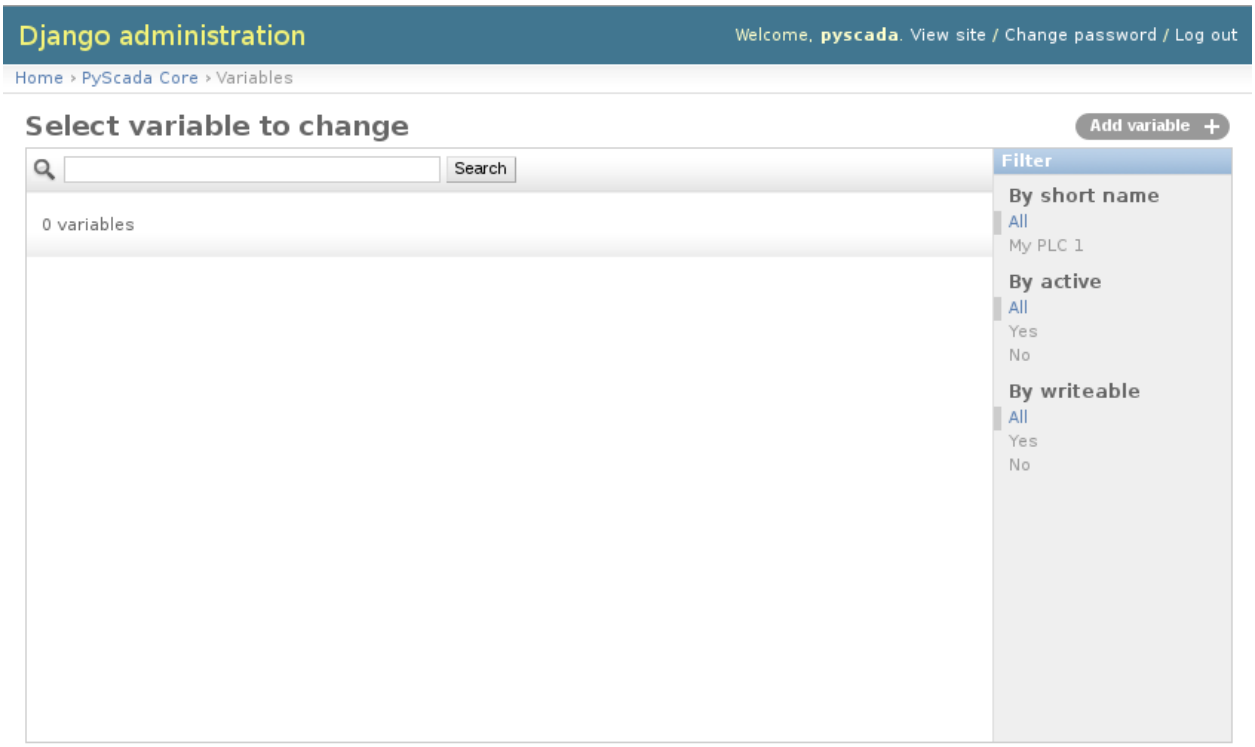
You will see a empty list. Click on *add modbus device* in the upper right corner to add a new device (e.g. a modbus device).



For Modbus IP only the IP-Address, the port of the modbus server and the *unit id* are required.

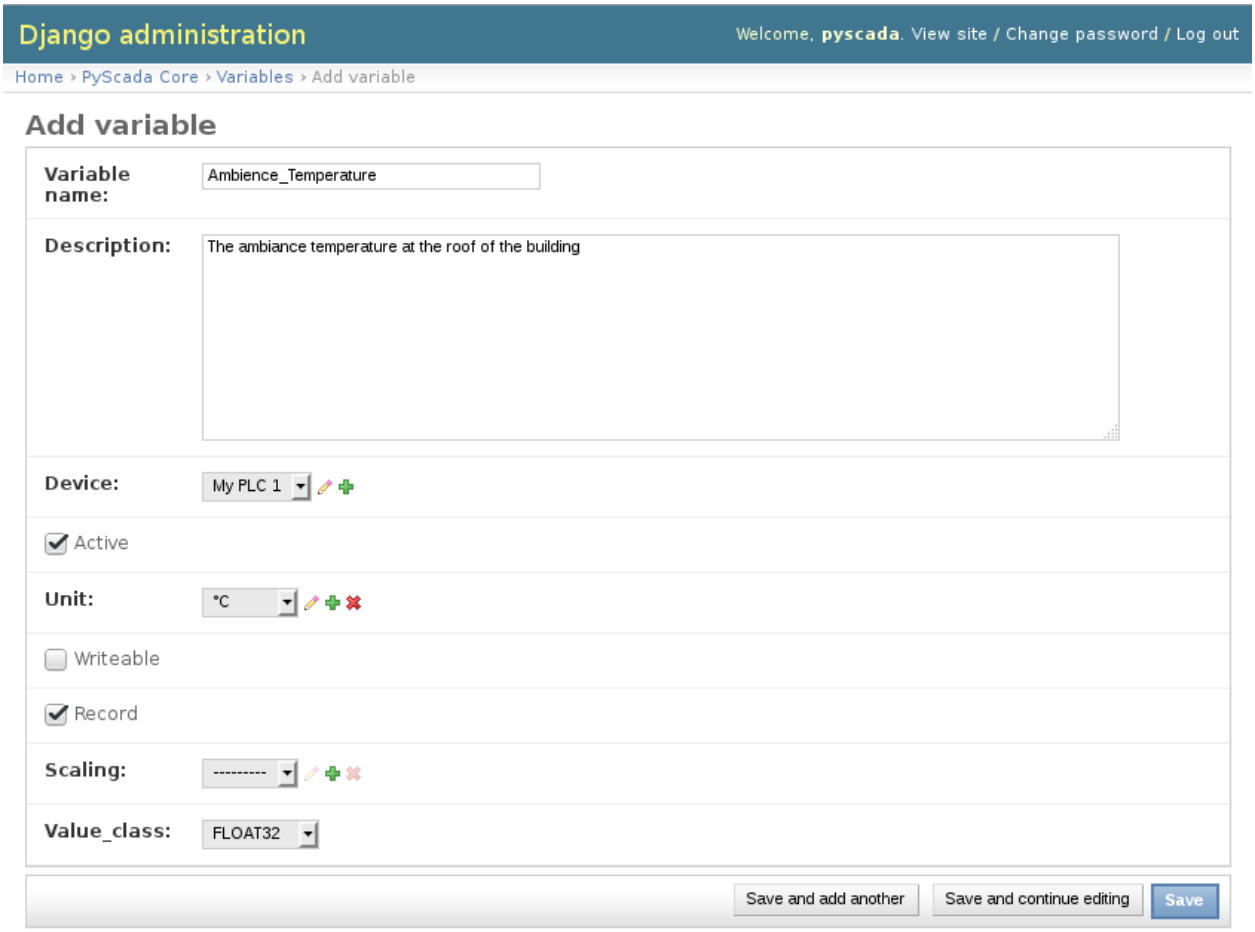
10.2 Add a new Variable

Adding a new variable/data point first add a new *Variable* entry to *Variables* table in the *PyScada Core* section of the admin panel, to do so click *add variable* in the upper right corner.

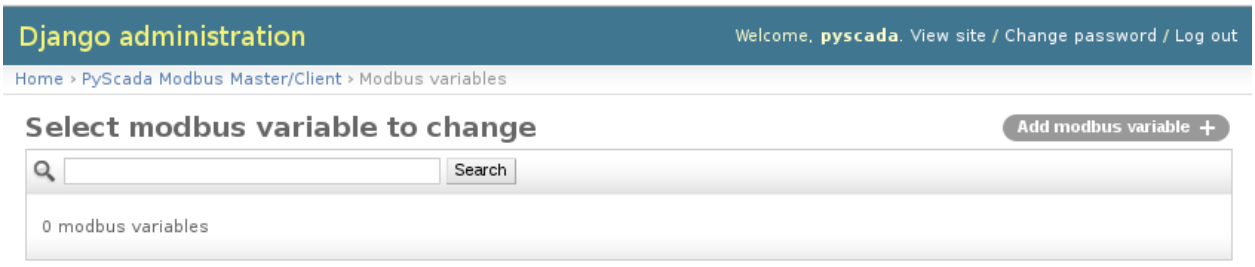


A Variable has a name and a description, assign the Variable to a Device and select a Unit of measurement (*TODO* add description off adding a new unit), activate writable if the value should be changed from the HMI, if the value has to be scaled in order to be displayed right select the right scaling (*TODO* add description for adding a scaling).

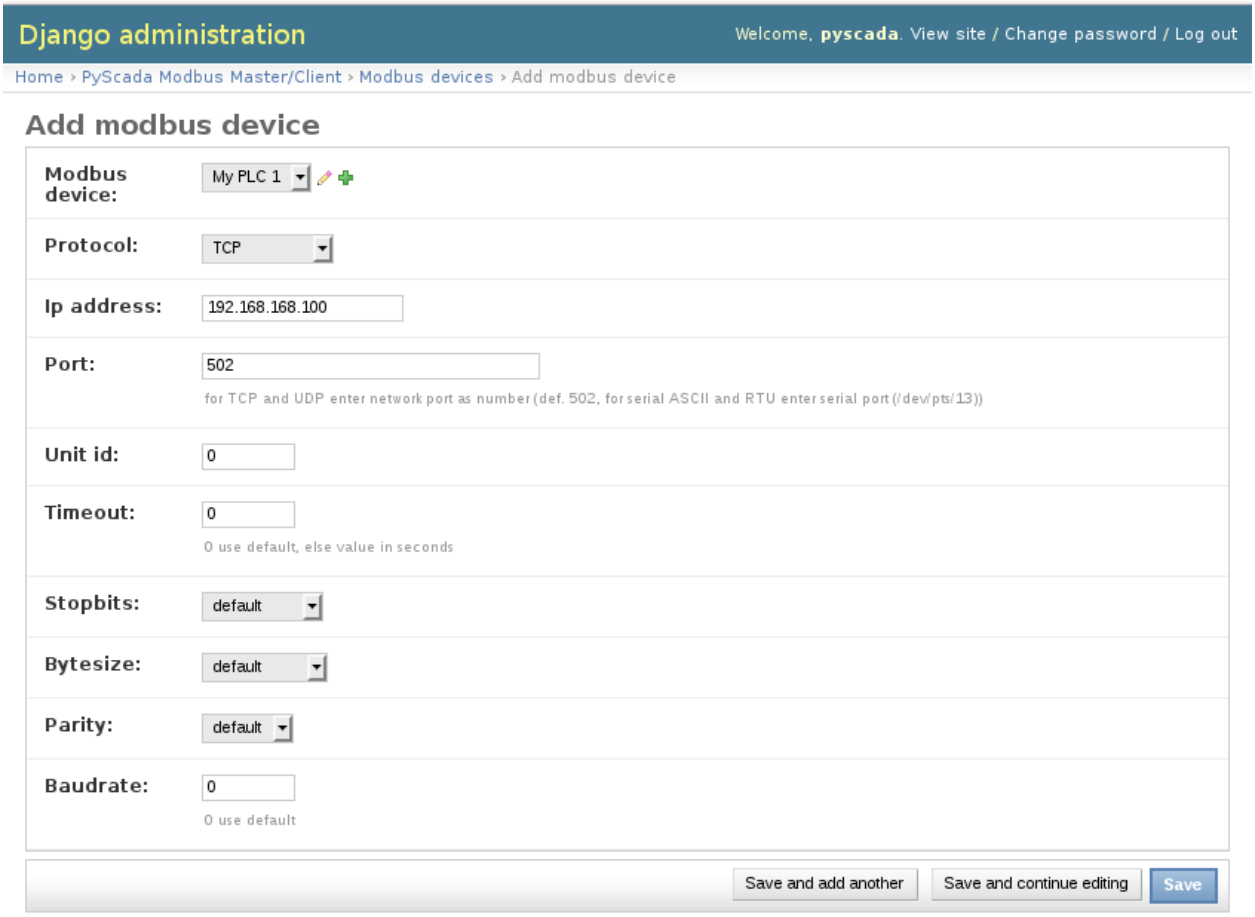
The *value_class* is the data type in witch the value is represented on the Device (*TODO* add example). The *cov_increment* is the amount of change of the value to be stored in the database.



We assume we want to add a modbus variable. To add the information about the register and the function code, add a new modbus variable, to do so open the *Modbus variables* table from the *PyScada Modbus Master/Client* section and click *add modbus variable* in the upper right corner.

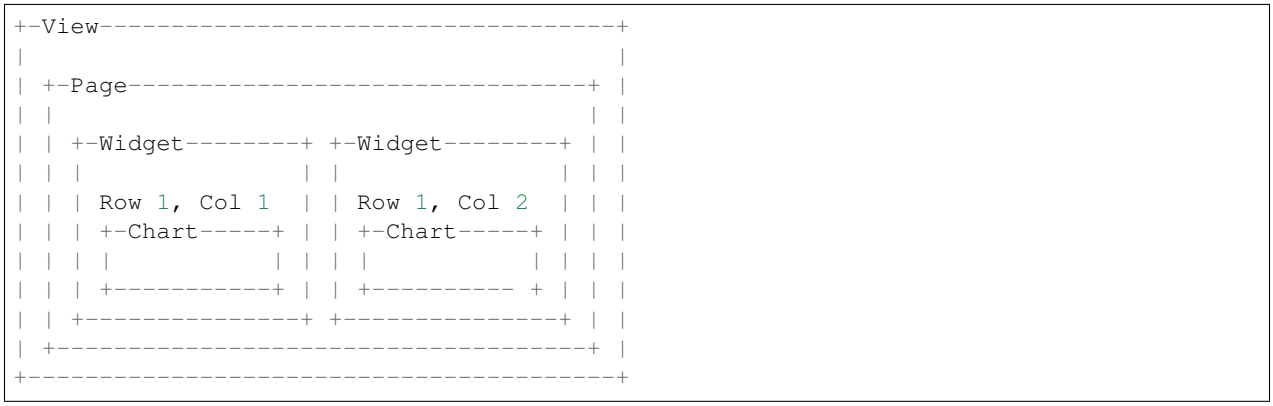


Select the Variable to assign by clicking on the magnifier symbol or writing the id of the Variable. Add the register address of the lowes word (a FLOAT32 variable on two 16 bit registers 1234 and 1235, 1234 has to be selected), for boolen values provide the address of the bit. For the *Function code read* the function code for reading the value from the modbus server has to be selected, for writing values to the server the coresponding function code will be selected automaticly (*TODO* function code table).



10.3 short instructions for the HMI

In the Backend HMI Section (http://IP/admin/hmi/): 1. Charts, add a new Chart 2. Page, add a Page 3. Widget, add a Widget, select under Page the page you added in 2. and under Chart the Chart from 1., a widget controls the position of every element on a Page. 4. View, add a View and select the page from 2. 5. GroupDisplayPermissions, add a new GroupDisplayPermission, (if nessesary add a new Group and add your User to that Group, select all items you created in 1. to 4. 6. open http://IP/, you should see the new View, if the DAQ is running and there is Data already in the DB, you should see the last 2 Houers of Data and the curennt Data.



10.4 Add a new View

to be continued...

10.5 Add a new Page

to be continued...

10.6 Add a new Chart

to be continued...

10.7 Add a new Control Panel

to be continued...

CHAPTER 11

Installation

add the following line to the urls.py:

```
url(r'^', include('pyscada.phant.urls')),
```

CHAPTER 12

Indices and tables

- genindex
- modindex
- search