Project Installation

This page describes the installation phase of the Pyros software

Project Development page : [[Project Development]]

HOWTO Format Redmine Wiki : http://www.redmine.org/projects/redmine/wiki/FrRedmineWikiFormatting

---

DATABASE SCHEMA (v0.2.2)

{{thumbnail(PYROS_DBM_V0_2_2.png, size=2000, title=Pyros data model)}}

COMPATIBLE PLATFORMS (TESTED)

This software is targetted first for Linux CentOS 7 (+ Fedora and Ubuntu), but also for Mac OS X and Windows.
All these systems should run Python 3 (3.5+)

Pyros has been tested on these platforms:

- CentOS 7.1 (with Python 3.4)
- Linux Mint 17.2 (== Ubuntu 14.04.3) (with Python 3.5)
- Ubuntu 16.04 (with python 3.5.2)
- Mac OS 10.11.6 (with Python 3.5)
- Windows 10 (with Python 3.5)

I - PREREQUISITES

Pyros needs some prerequisites :

- Python 3.5+
- RabbitMQ
- [optional] a Mysql Database server (but you can use Sqlite instead, with no installation)

Install Python3 (3.5+) + pip + virtualenv

If Python 3 is already installed on your computer, skip this section

- Linux (CentOS 7.1, main target): (python35 not yet available as rpm)

```
$ sudo yum update yum
$ sudo yum update kernel
$ sudo yum update

$ sudo yum install python34
$ python3.4 -V
Python 3.4.3

$ sudo yum install python34-devel
(needed for python package mysqlclient)

$ sudo yum update python-setuptools
$ easy_install --version
setuptools 0.9.8
$ sudo easy_install pip
$ pip --version
pip 8.1.1 from /usr/lib/python2.7/site-packages/pip-8.1.1-py2.7.egg (python 2.7)
```

```
$ sudo pip install virtualenv
$ virtualenv --version
15.0.1
```

Necessary for "lxml" python package:

```
$ sudo yum install libxml2 libxml2-devel
$ sudo yum install libxslt libxslt-devel
```

- Linux (Ubuntu) :

```
$ sudo add-apt-repository ppa:fkrull/deadsnakes
$ sudo apt-get update
$ sudo apt-get install python3.5
$ sudo apt-get install libxml2-dev
$ sudo apt-get install libxslt-dev
$ sudo apt-get install python-pip

$ sudo apt-get install python3.5-dev
```

(needed for python package mysqlclient)

```
$ sudo pip install virtualenv
```

- Mac OS X :
  - From binary (Mac OS native application)
    (TODO:)
  - From MacPort
    - a) install macport
      cf https://www.macports.org/install.php
    - b) install the "port" python35

```
$ sudo port install python35
```

- Windows (tested with Windows 7 & 10) :

Go to https://www.python.org/downloads/windows/ , choose the wanted version
On the wanted version’s page, download Windows x86 executable installer

Run the executable
 * On the first page, check "Add python3.5 to PATH"
 * Choose "Install now" option

Open cmd (windows + R, cmd) :

```
 $ python -m pip install --upgrade pip
 $ pip install virtualenv
```

Install a database server (only if necessary)

If the MySql database server is already installed on your computer, skip this section

By default, Pyros uses Mysql, but this implies that you have to install the Mysql database server...

Thus, to make things easier, avoid Mysql installation by using Sqlite instead as the database server (which will need no installation at all) :

=> **For this, just edit the file PYROS/src/pyros/settings.py and set MYSQL variable to False, and that's it. You can go to next section**

Now, if you really want to use Mysql (which is the default), you will need to install it (only if not already installed), so keep reading.

(Skip this if you are using Sqlite instead of MySql)

- Linux Ubuntu

First, update your system:

```
$ sudo apt-get update

$ sudo apt-get install mysql-server
$ sudo apt-get install mysql-client

$ sudo apt-get install libmysqlclient-dev
```

(needed for python package mysqlclient)

- Linux CentOS : cf https://www.howtoforge.com/apache_php_mysql_on_centos_7_lamp#-installing-mysql-

First, update your system:

```
$ sudo yum update yum
$ sudo yum update kernel
$ sudo yum update

$ sudo yum install mariadb-server
$ sudo yum install mariadb

$ sudo yum install mariadb-devel
```

(needed for python package mysqlclient)

```
$ sudo systemctl start mariadb.service

$ sudo systemctl enable mariadb.service
=> Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to
/usr/lib/systemd/system/mariadb.service.

$ sudo mysql_secure_installation
```

- Mac OS X
  Install XAMPP
  (but you could also use the pre-installed Mac OS MySql, or install MySql from macport)
  cf https://www.apachefriends.org/fr/index.html

- Windows (tested with Windows 7 & 10)

Download and install the newest version on https://dev.mysql.com/downloads/installer/

Once installed, launch MySQL Installer. Clic on 'Add...' on the right.
In MySQLServers section, choose the newest, then clic on next.
Install and configure the server (just follow the installation guide).

Then launch mysql (via the Windows menu).

Install RabbitMQ

RabbitMQ is a message queue server used by Celery to handle tasks queues.
It uses the amqp protocol to manage queue messages.

- Ubuntu (the server is automatically started) :

```
$ sudo apt-get install rabbitmq-server
```

Get status:

```
$ sudo invoke-rc.d rabbitmq-server status
```

Stop:

```
$ sudo invoke-rc.d rabbitmq-server stop
```

Start:

```
$ sudo invoke-rc.d rabbitmq-server start
```

- CentOS :

```
$ sudo yum install rabbitmq-server
```

```
Installation :
 rabbitmq-server                       noarch                3.3.5-17.el7
Installation pour dépendances :
 erlang-asn1                         x86_64                R16B-03.16.el7
```

Get status:

```
(CentOS7) $ sudo rabbitmqctl status
(older CentOS) $ sudo /sbin/service rabbitmq-server status
```

Stop:

```
(CentOS7) $ sudo systemctl stop rabbitmq-server
(older CentOS) $ sudo /sbin/service rabbitmq-server stop
```

Start:

```
(CentOS7) $ sudo systemctl start rabbitmq-server
(older CentOS) $ sudo /sbin/service rabbitmq-server start
```

- MacOS :

With MacPort:

```
$ sudo port install rabbitmq-server
--->  Installing erlang @18.2.1_1+hipe+ssl
...
--->  Installing rabbitmq-server @3.5.7_0
--->  Activating rabbitmq-server @3.5.7_0
...
```

Get status:

```
$ sudo rabbitmqctl status
```

To start rabbitmq :

```
$ sudo rabbitmq-server
```

To stop rabbitmq :

```
$ sudo rabbitmqctl stop
```

- Windows :
  - Take the wanted Erlang version at http://www.erlang.org/downloads and install it (required)
  - Take the wanted RabbitMQ version at https://www.rabbitmq.com/install-windows.html and install it. Then the server will run automatically

II - GET THE PYROS SOFTWARE

Authenticate to https://gitlab.irap.omp.eu/epallier/pyros

In order to get this software, you must first authenticate on the IRAP gitlab.

For this, just go to https://gitlab.irap.omp.eu/epallier/pyros
and either sign in with your LDAP account (if you are from IRAP),
or register via the "Sign up" form.

Software structure

See [[Project_structure|Project Structure]] page.

Browse the code (from gitlab)

https://gitlab.irap.omp.eu/epallier/pyros/tree/master

You can download it (static version, not synchronized)

You can download the current version of the software.

Go to https://gitlab.irap.omp.eu/epallier/pyros/tree/master

Click on "Download zip" on the up right hand corner.

Double-click on it to unzip it.

You should get a "pyros.git" folder.

In this documentation, this software folder will be referenced as "PYROS".

(you can rename "pyros.git" as "PYROS" if you want : "mv pyros.git PYROS")

(Windows developers only) : Get git for Windows

- Download git at https://git-scm.com/download/win
- Run setup (keep default configurations)
- Once installed, open cmd

```
$ git config --global http.sslVerify false
```

You can now use your git from the cmd or the graphic client !

(Developers only) Get it from GIT (dynamic version, synchronized)

(If you are using Windows, you must first get the git software, see below "Get git for Windows")

By getting the software from git, you will get a dynamically synchronized version,
which means that you will be able to update your version as soon as a new version is available
(with the command : "git pull").

- From Eclipse : See below, section "NOTES FOR ECLIPSE USERS"

- From the terminal

```
git clone https://gitlab.irap.omp.eu/epallier/pyros.git PYROS
```

(or also : git clone git@gitlab.irap.omp.eu:epallier/pyros.git PYROS)

If you ever get this error message :

```
fatal: unable to access 'https://gitlab.irap.omp.eu/epallier/pyros.git/': Peer's certificate issuer has been marked as not trusted by the user.
```

Then, type this command (and then run again the git clone command):

```
$ git config --global http.sslVerify false
```

(Also, the first time you get the project, it will ask you for a login and password)

This creates a PYROS/ folder containing the project (with a .git/ subfolder for synchronization with the git repository)

III - INSTALLATION OF NEEDED PROJECT DEPENDENCIES

Install Comet

Comet is needed as a broker to receive and send VOEvents (https://github.com/jdswinbank/Comet/tree/py3)
You MUST have your virtualenv activated (source venv_py3_pyros/bin/activate in your 'private/' directory)
Documentation is available here : http://comet.readthedocs.io/en/stable/installation.html
(see also http://voevent.readthedocs.io/en/latest/setup.html)

- Ubuntu :

```
# You can do this anywhere on your computer
$ git clone https://github.com/jdswinbank/Comet.git
$ cd Comet
$ (sudo ?) python setup.py install
$ sudo apt-get install python-lxml

# Now you can test it, no failure are expected, please contact the dev team for help
$ twistd comet --help
$ trial comet
All tests should pass
```

- MacOS :

Idem Ubuntu

- Windows :

TODO:

Install the other project dependencies and the database (within a Python virtual environment)

Go into the PYROS/install/ folder and Run the install.py script with the parameter install.

(You will be prompted to enter a username, an email, and a password for the superuser)

- Mac, Linux and Windows:

```
python3 install.py install
```

If something goes wrong, install manually each package (see section "MANUAL INSTALLATION OF PYTHON PACKAGES, ONE BY ONE")

(Eclipse only) Set the Python3 virtual environment as the project interpreter

See below, in section "NOTES FOR ECLIPSE USERS"

IV - TEST

(Tests are classes declared in all apps' test.py file. The test classes inherit from django.test.TestCase)

You can run the tests suite, just to be sure that the software is well installed.

From Terminal

```
$ cd PYROS/

Be sure that at least unit tests pass:
$ python3 pyros.py unittest

If unit tests pass, then run ALL tests:
$ python3 pyros.py test_all

NB: if tests do not pass, be sure that Rabbitmq is launched:
To start rabbitmq on MacOS :
$ sudo rabbitmq-server
```

From Eclipse

Right-clic on the project / Django / Run Django tests

(for more information, see below, in section "NOTES FOR ECLIPSE USERS")

Custom commands

```
$ [./manage.py] test app.tests                          # Run tests for the application 'app'
$ [./manage.py] test app.tests.ModelTests               # Run test methods declared in the class app.tests.ModelTests
$ [./manage.py] test app.tests.ModelTests.test_method    # Only run the method test_method declared in app.tests.ModelTests
```

V - RUN

Before launching Pyros, check that all the pre-requisites are running :

- RabbitMQ:

  ```
  $ systemctl status rabbitmq-server
  ```

  If inactive, start it:
  ```
  $ sudo systemctl start rabbitmq-server
  ```

  (To stop it, use "stop" instead of "start")

- Mysql (only if you are not using Sqlite):

  ```
  $ systemctl status mysqld
  ```

  If inactive, start it:
  ```
  $ sudo systemctl start mysqld
  ```

  (To stop it, use "stop" instead of "start")

- From the terminal

  ```
  $ cd PYROS/
  ```

  ```
  $ python3 pyros.py start
  ```

  (To check that this service is actually running, type "$ netstat -an |grep 8000" and you should get "tcp   0   0 127.0.0.1:8000   0.0.0.0:*   LISTEN")

  (Use instead "python manage.py runserver YOUR_IP_ADDRESS:8000" if you want to access the website from another pc with http://YOUR_IP_ADDRESS:8000) (from within a virtual env, and from src/)

Now, go to "http://localhost:8000/admin" in your browser

(and login as "pyros" with the password 'DjangoPyros')

(Ctrl-c to stop)

- From Eclipse : see below, in section "NOTES FOR ECLIPSE USERS"

NOTES FOR ECLIPSE USERS

0) Install Eclipse (if necessary)

Get the Eclipse version for Php

Install the plug-in pydev (via install new software, add http://pydev.org/updates)

Install the django template editor (via install new software, add http://eclipse.kacprzak.org/updates)

1) Deactivate CA certificate verification

Window -> Preferences -> Team -> git -> configuration -> Add entry

Key = http.sslVerify
Value = false

2) Fetch the project

File/Import project from git

Select repository source: Clone URI: https://gitlab.irap.omp.eu/epallier/pyros.git

Directory:
par défaut, il propose : /Users/epallier/git/pyros
mais on peut le mettre ailleurs (c'est ce que j'ai fait)

initial branch: master
remote name: origin

Import as general project

Project name: PYROS

3) Configure the project

The project is created. Now, you have to set it as a «PyDev » and a « Django » project.

clic droit sur le projet / PyDev / set as a PyDev project

clic droit sur le projet / PyDev / set as a Django project

clic droit sur le projet : on doit maintenant avoir un sous-menu Django

clic droit sur le dossier src / PyDev / set as source folder (add to PYTHONPATH)

clic droit sur projet / Properties / Pydev-Django :
- **Django manage.py : src/manage.py**
- **Django settings module : pyros.settings**

4) Set Code style

Window/Preferences : Pydev / Editor
- Auto Imports : uncheck « Do auto import »
- Code style:
  - Locals … : camelCase
  - Methods : camelCase() with first lower
- Code style / Code Formatter: activer « use autopep8.py for code formatting »
- Tabs : Tab length : 4

5) Now, once the Python3 virtual environment is installed (see above),
set it in Eclipse as the project interpreter:

Right clic on the project / Properties / PyDev - Interpreter/Grammar :
Grammar version : 3

Interpreter : click on « click here to configure an interpreter not listed »

click on « New... » :
- Interpreter name : venv_py35_pyros

- Interpreter executable : click on « Browse »
  Select your virtualenv python35 executable (private/venv_py35_pyros/bin/python3.5)
  click OK

Now, a last thing to do so that Pydev finds Django:

In the Python interpreters list, select your virtual env (venv_py35_pyros)

Click on the tab "Libraries"

clic on 'New folder', then select your virtualenv's lib/python3.5/site-packages folder

OK

Interpreter: select now venv_py35_pyros from the list

OK

6) Test

- Right-clic on the project / Django / Run Django tests

- Right-clic on the project / Django / Custom command / ...

7) Run

Right clic on project -> Django/Custom command/runserver

Error on Ubuntu :

```
Traceback (most recent call last):
  File "/home/carens_p/git/pyros/src/manage.py", line 8, in <module>
    from django.core.management import execute_from_command_line
ImportError: No module named 'django'
```

Project -> Properties -> Pydev interpreter -> Clic here to configure an interpreter not listed

select venv_py35_pyros

In Libraries, clic on 'New folder', then select your virtualenv's lib/python3.5/site-packages folder.

OK -> re-select venv_py35_pyros in 'Interpreter' dropdown menu

Then right clic on project -> Django/Custom command/runserver

check http://localhost:8000/

NOTES FOR PYCHARM USERS

1) Install Pycharm

2) import pyros project

3) Mark the src directory and simulators directory as source root directories

4) Go in file -> settings (CTRL + ALT + S) -> Project : Pyros -> Project Interpreter
Add an interpreter which is the one from your virtual environment : Add Local -> find the python 3 binary in your virtualenv

5) Go in Language & Frameworks -> Django and set the django project root / Settings (pyros/settings.py) / Manage script

---

MANUAL INSTALLATION OF PYTHON PACKAGES ONE BY ONE

Follow these steps only if the previous guided and nearly automatic installation did not work for you

(Only if using Mysql) Create the database "pyros" and the pyros user

Only if you are using Mysql, you need to create an empty database "pyros"
(which will be filled automatically by django)

```
$ mysql -u root -p
(enter your root password)

$ mysql> create database if not exists pyros;
```

The user creation depends on your MySQL version :

- 5.7 and above :

```
$ mysql> DROP USER IF EXISTS pyros;
$ mysql> CREATE USER 'pyros' IDENTIFIED BY 'DjangoPyros';
$ mysql> GRANT ALL PRIVILEGES ON pyros.* TO pyros;
```

- under 5.7 :

```
$ mysql> GRANT USAGE ON *.* TO 'pyros';
$ mysql> DROP USER 'pyros';
$ mysql> CREATE USER 'pyros' IDENTIFIED BY 'DjangoPyros';
$ mysql> GRANT ALL PRIVILEGES ON pyros.* TO pyros;
```

If none of these solution work, check on the internet to create a user named pyros with the password DjangoPyros.

Create a Python3 virtual environment dedicated to the project (inside the project folder)

```
$ mkdir private/

$ cd private/

$ which python3.5 ("where python" for windows)
/opt/local/bin/python3.5

$ virtualenv venv_py35_pyros -p /opt/local/bin/python3.5
=> creates a venv_py35_pyros/ folder inside PYROS/private/
```

Activate the python virtual environment (from inside the project)

```
$ pwd
.../PYROS/private

$ source ./venv_py35_pyros/bin/activate (venv_py35_pyros/Scripts/activate on Windows)
```

Install needed python packages

Check that the virtual environment is activated

```
$ python -V
Python 3...

$ which pip
.../PYROS/venv_py35_pyros/bin/pip
```

Upgrade pip to last version available:

```
$ pip install --upgrade pip
Collecting pip
  Downloading pip-8.1.1-py2.py3-none-any.whl (1.2MB)
Installing collected packages: pip
  Found existing installation: pip 7.1.2
    Uninstalling pip-7.1.2:
      Successfully uninstalled pip-7.1.2
Successfully installed pip-8.1.1

Upgrade wheel to last version available:
$ pip install --upgrade wheel
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
Installing collected packages: wheel
  Found existing installation: wheel 0.24.0
    Uninstalling wheel-0.24.0:
      Successfully uninstalled wheel-0.24.0
Successfully installed wheel-0.29.0
```

Go into the install/ folder:

```
$ cd .../PYROS/install/
```

Install all the needed python packages at once:

```
$ pip install -r REQUIREMENTS.txt
```

If something goes wrong, install them one by one:

- **Install Django** :

```
$ pip install django
Collecting django
```

```
  Downloading Django-1.9.4-py2.py3-none-any.whl (6.6MB)
Installing collected packages: django
Successfully installed django-1.9.4

$ pip install django-admin-tools
Collecting django-admin-tools
  Downloading django_admin_tools-0.7.2-py2.py3-none-any.whl (289kB)
Installing collected packages: django-admin-tools
Successfully installed django-admin-tools-0.7.2

$ pip install django-debug-toolbar
Collecting django-debug-toolbar
  Downloading django_debug_toolbar-1.4-py2.py3-none-any.whl (212kB)
Requirement already satisfied (use --upgrade to upgrade): Django>=1.7 in ./venv_py35_pyros/lib/python3.5/site-packages (from django-debug-toolbar)
Collecting sqlparse (from django-debug-toolbar)
  Downloading sqlparse-0.1.19.tar.gz (58kB)
Building wheels for collected packages: sqlparse
  Running setup.py bdist_wheel for sqlparse ... done
  Stored in directory:
/Users/epallier/Library/Caches/pip/wheels/7b/d4/72/6011bb100dd5fc213164e4bbee13d4e03261dd54ce6a5de6b8
Successfully built sqlparse
Installing collected packages: sqlparse, django-debug-toolbar
Successfully installed django-debug-toolbar-1.4 sqlparse-0.1.19

$ pip install django-extensions
Collecting django-extensions
  Downloading django_extensions-1.6.1-py2.py3-none-any.whl (202kB)
Collecting six>=1.2 (from django-extensions)
  Downloading six-1.10.0-py2.py3-none-any.whl
Installing collected packages: six, django-extensions
Successfully installed django-extensions-1.6.1 six-1.10.0

$ pip install django-suit
Collecting django-suit
  Downloading django-suit-0.2.18.tar.gz (587kB)
Building wheels for collected packages: django-suit
  Running setup.py bdist_wheel for django-suit ... done
  Stored in directory:
/Users/epallier/Library/Caches/pip/wheels/12/8b/9a/e02ab0ad9229881638aa040d47d77c8f562999533811927d41
Successfully built django-suit
Installing collected packages: django-suit
Successfully installed django-suit-0.2.18
```

- **Install the django boostrap css package** :

```
$ pip install django-bootstrap3
```

(==> 'bootstrap3' is then to be added as an application in settings.py -> INSTALLED_APPS)

- **Install the web application server gunicorn (will be used in production instead of the dev django web server)** :

```
$ pip install gunicorn
Collecting gunicorn
  Downloading gunicorn-19.4.5-py2.py3-none-any.whl (112kB)
Installing collected packages: gunicorn
Successfully installed gunicorn-19.4.5
```

- **Install the python mysql client** (not needed if you want to use sqlite):

```
$ pip install mysqlclient
```

...

- => If issue under Mac OS X:

```
$ pip install mysqlclient
Collecting mysqlclient
  Downloading mysqlclient-1.3.7.tar.gz (79kB)
Building wheels for collected packages: mysqlclient
  Running setup.py bdist_wheel for mysqlclient ... error
…
  ----------------------------------------
  Failed building wheel for mysqlclient
  Running setup.py clean for mysqlclient
Failed to build mysqlclient
Installing collected packages: mysqlclient
  Running setup.py install for mysqlclient ... done
Successfully installed mysqlclient-1.3.7
```

BOUH !!!

=> Need to upgrade wheel:

```
$ pip install --upgrade wheel
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
Installing collected packages: wheel
  Found existing installation: wheel 0.24.0
    Uninstalling wheel-0.24.0:
      Successfully uninstalled wheel-0.24.0
Successfully installed wheel-0.29.0
```

YES !!!

Only if necessary, you can reinstall mysqlclient:

```
$ pip uninstall mysqlclient
$ pip install mysqlclient
Collecting mysqlclient
  Using cached mysqlclient-1.3.7.tar.gz
Building wheels for collected packages: mysqlclient
  Running setup.py bdist_wheel for mysqlclient ... done
  Stored in directory:
/Users/epallier/Library/Caches/pip/wheels/9b/06/50/d11418c26cf8f2156b13d4363b5afde8e7e75ebb8540d0228d
Successfully built mysqlclient
Installing collected packages: mysqlclient
Successfully installed mysqlclient-1.3.7
```

- => If issue under Windows

Same message as the issue for Mac.

=> Need to install wheel manually :

Go to http://www.lfd.uci.edu/~gohlke/pythonlibs/#mysqlclient to download the newest mysqlclient wheel

```
$ pip install path\to\mysqlclient\wheel
```

(No need to redo "pip install mysqlclient")

- **Install the julian day converter** :

```
$ pip install jdcal
```

- **Install Celery and dependencies** :

```
$ pip install celery
$ pip install django-celery
$ pip install Twisted==16.0.0
```

- **Install django test without migrations (compulsory to use the prod DB for tests)** :

```
$ pip install django-test-without-migrations==0.4
```

- **Install voevent parser** :

```
$ pip install voevent-parse==0.9.5
```

- **Install other dependencies (useful ? TBC)** :

```
$ pip install amqplib==1.0.2
$ pip install pluggy==0.3.1
$ pip install py==1.4.31
```

Files

PYROS_PDM_v021.png	306 KB	03/24/2016	Etienne Pallier
PYROS_DBM_V0_2_2.png	304 KB	08/11/2016	Etienne Pallier