

## Project Installation

---

Project Development page : [[Project Development]]

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

---

### I - DATABASE SCHEMA (v0.2.1)

---

{{thumbnail(PYROS\_PDM\_v021.png, size=300, title=Pyros data model)}}

---

### II - Get the project (from gitlab)

---

#### Browse the code from gitlab

---

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

#### Project structure

---

Once you have fetched the project, you should obtain this structure:

```
PYROS/  
  install/  
  REQUIREMENTS.txt  
  INSTALLATION.txt  
  public/  
  static/  
  src/  
  manage.py  
  pyros/  
  __init__.py  
  __pycache__  
  settings.py  
  urls.py  
  wsgi.py  
  pyrosapp/  
  __init__.py  
  admin.py  
  apps.py  
  migrations  
  models.py  
  tests.py  
  views.py
```

#### 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)
```

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

*If you just wanted a static copy of the project (without synchronization), just remove the .git/ folder:*

```
$ rm -r .git/
```

## From Eclipse

---

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>)

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

(

il faudra aussi faire un peu plus loin :

[https://projects.irap.omp.eu/projects/pyros/wiki/Project\\_Installation#Eclipse-only-Set-the-Python3-virtual-environment-as-the-project-interpretter](https://projects.irap.omp.eu/projects/pyros/wiki/Project_Installation#Eclipse-only-Set-the-Python3-virtual-environment-as-the-project-interpretter)

)

**For easier installation, you can use Sqlite instead of Mysql**

---

By default, Mysql will be used, but then you will need to install the Mysql database server.

Thus, in order to skip Mysql installation, use Sqlite instead as the database server (which will need no installation at all) :

Edit pyros/settings.py and just set MYSQL variable to False, and that's it.

### III - INSTALLATION

---

#### Install MySql (only if necessary)

---

**Skip this step if you are using Sqlite instead of MySql**

- Linux Ubuntu

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

- Linux CentOS

TODO:

```
$ sudo yum install mysql
```

...

- Mac OS X

Install XAMPP

(but you could also use the pre-installed Mac OS MySql)

TODO:

- Windows

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 Python3.5 (only if necessary)

---

- Mac OS X :

1) Installer MacPort  
(TODO: doc)

2) Installer le "port" python35  
\$ sudo port install python35

- Linux (Ubuntu) :

```
$ sudo add-apt-repository ppa:fkruhl/deadsnakes
$ sudo apt-get update
$ sudo apt-get install python3.5
```

```
$ sudo pip install virtualenv
```

- Windows 7 :

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
```

- Windows 10 :

TODO:

---

## 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-3.5 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)
```

```
$ python -V
Python 3.5.1
```

```
$ 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
```

---

## Install the needed Python packages (from within the virtual environment)

---

First, be sure that the virtual environment is activated:

```
$ python -V
Python 3.5.1
```

### - Automatic Installation of all packages

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

### - Or, manual installation of each package

#### - 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 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:**

```
$ pip install mysqlclient
...
```

- => 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 !!!

```
$ 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
```

```
$ 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

YES !!!

- => Issues under Ubuntu:

```
$ pip install mysqlclient
Collecting mysqlclient
  Downloading mysqlclient-1.3.7.tar.gz (79kB)
  Complete output from command python setup.py egg_info:
  /bin/sh: 1: mysql_config: not found
  Traceback (most recent call last):
    File "<string>", line 1, in <module>
  [...]
-----
Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-q6j4inuz/mysqlclient/
```

BOUH !!!

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

```
$ 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 ... error
...
_mysql.c:40:20: fatal error: Python.h: No such file or directory
  #include "Python.h"
                ^
compilation terminated.
error: command 'x86_64-linux-gnu-gcc' failed with exit status 1
```

```
-----
Command "/home/carens_p/pyros/venv_py35_pyros/bin/python3.5 -u -c "import setuptools,
tokenize;__file__='/tmp/pip-build-k3klv92j/mysqlclient/setup.py';exec(compile(getattr(tokenize, 'open',
open)(__file__).read().replace('\r\n', '\n'), __file__, 'exec'))" install --record /tmp/pip-gz242xxs-record/install-record.txt
--single-version-externally-managed --compile --install-headers
/home/carens_p/pyros/venv_py35_pyros/include/site/python3.5/mysqlclient" failed with error code 1 in
/tmp/pip-build-k3klv92j/mysqlclient/
```

BOUH !!!

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

```
$ pip install mysqlclient
```

YES !!!

- => Issue under Windows

Same message as the issue for Mac.

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"

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

---

cliquez sur projet / Properties / PyDev - Interpreter/Grammar :  
Grammar version : 3

Interpreter : cliquez sur « cliquez ici pour configurer un interpréteur non répertorié »

cliquez sur « Nouveau » :

- Nom de l'interpréteur : venv\_py35\_pyros
- Exécutable de l'interpréteur : cliquez sur « Parcourir »
  - Sélectionnez votre private/venv\_py35\_pyros/bin/python3.5 exécutable
  - cliquez OK
  - cliquez OK à nouveau

Interpreter: sélectionnez maintenant venv\_py35\_pyros de la liste

---

## Run the project

---

TODO: plus de détails... (notamment sur localhost:8000/admin)

### From the terminal

---

```
$ cd src/  
  
$ ./manage.py runserver  
(or gunicorn pyros.wsgi)  
==> http://localhost:8000  
...  
...  
Ctrl-c
```

### From Eclipse

---

cliquez à droite sur projet -> Django/Custom command/runserver

Erreur sur 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'
```

Projet -> Properties -> Pydev interpreter -> Cliquez ici pour configurer un interpréteur non répertorié

sélectionnez venv\_py35\_pyros

Dans les bibliothèques, cliquez sur 'Nouveau dossier', puis sélectionnez votre virtualenv's lib/python3.5/site-packages dossier.

OK -> re-sélectionnez venv\_py35\_pyros dans le menu déroulant 'Interpreter'

Puis cliquez à droite sur projet -> Django/Custom command/runserver

vérifiez <http://localhost:8000/>

---

## IV - TEST

---

Les tests sont des classes déclarées dans tous les fichiers test.py des applications. Les classes de test héritent de django.test.TestCase



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

## From Terminal

---

```
$ cd src/
```

```
$ ./manage.py test
```

```
Ran 5 tests in 0.093s
```

```
OK
```

## From Eclipse:

---

- Right-click on the project / Django / Run Django tests
- Right-click on the project / Django / Custom command / ...

## 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
```

## Fichiers

---

PYROS\_PDM\_v021.png

306,365 ko

24/03/2016

Etienne Pallier