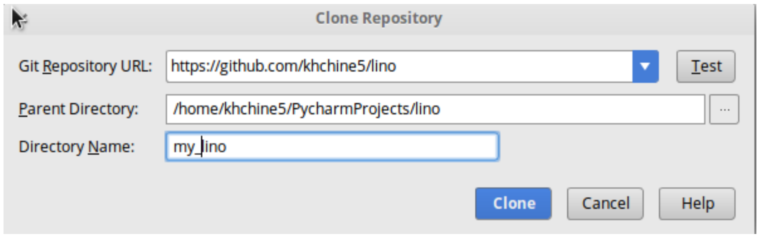
Lino-Framework.org

Discovering Lino using your debugger

In my daily work , I am using Pycharm as my favorite IDE .It is very efficient when debugging Python code and have many useful features. Let’s see who I use it to debug Lino base code to fix issues or add new code.
The next steps assume that you have Pycharm installed in your machine.

1) Creating a new Pycharm project

Lino code base lives in its official repository in Github. First of all , you need to clone lino project to your github account to be able to commit your change. This is done in the github website.
Once this is done , you need to fetch the code from your new repository and then install it within your virtual python environment.

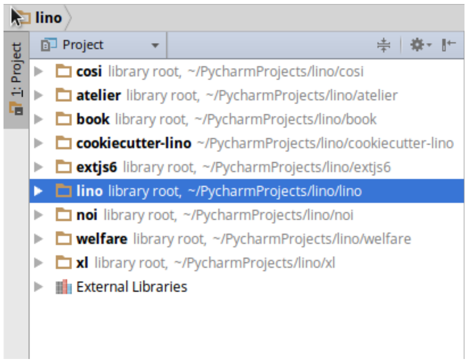

Clone Repository

Git Repository URL: https://github.com/khchine5/lino

Parent Directory: /home/khchine5/PycharmProjects/lino

Directory Name: my_lino

-Git Repository URL : the URL of your project.
-Parent Directory : Where your project will be copied in your machine.
After this, the project will be opened in Pycharm projects section

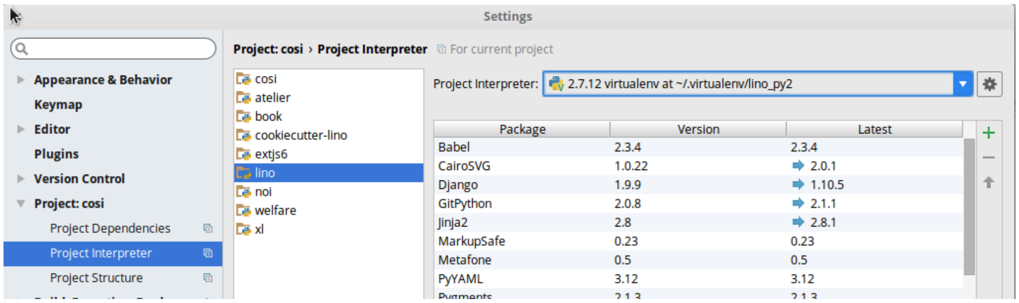

lino

Project

- cosi library root, ~/PycharmProjects/lino/cosi
- atelier library root, ~/PycharmProjects/lino/atelier
- book library root, ~/PycharmProjects/lino/book
- cookiecutter-lino ~/PycharmProjects/lino/cookiecutter-lino
- extjs6 library root, ~/PycharmProjects/lino/extjs6
- lino library root, ~/PycharmProjects/lino/lino
- noi library root, ~/PycharmProjects/lino/noi
- welfare library root, ~/PycharmProjects/lino/welfare
- xl library root, ~/PycharmProjects/lino/xl
- External Libraries

2)Add your virtual python environment to Pycharm

Open you Pycharm setting and search for project interpreter



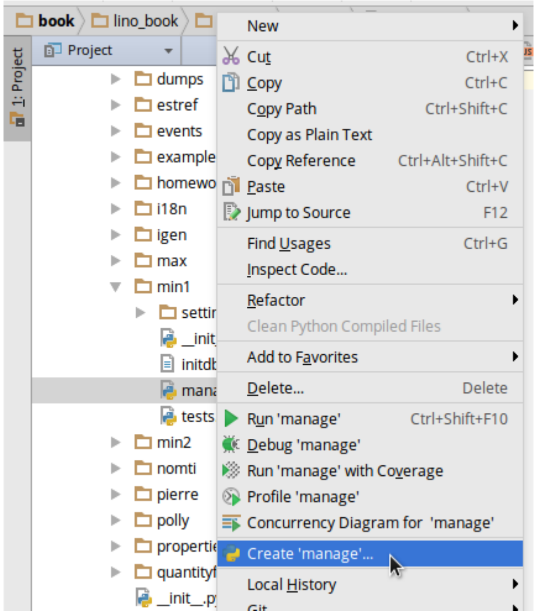
Project: cosi › Project Interpreter   For current project

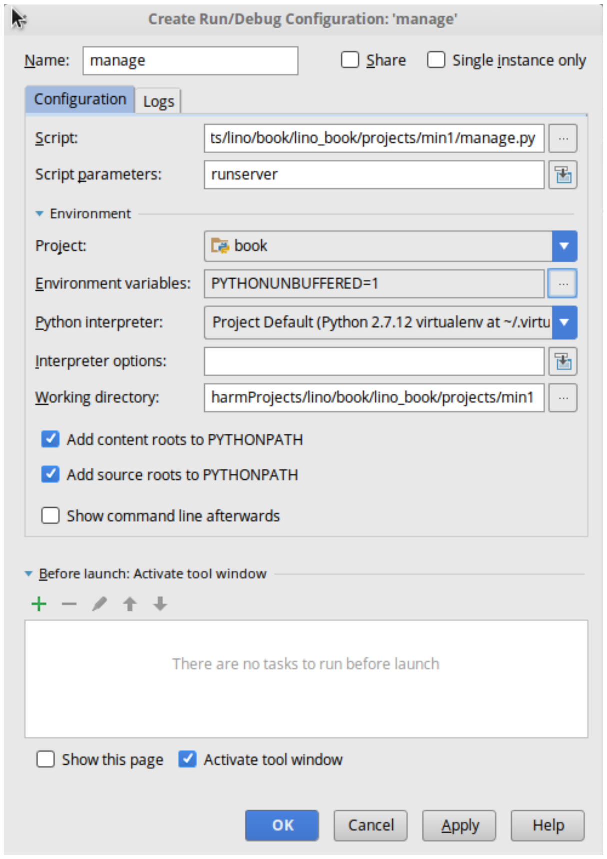
Project Interpreter: 2.7.12 virtualenv at ~/.virtualenv/lino_py2

Package	Version	Latest
Babel	2.3.4	2.3.4
CairoSVG	1.0.22	➡ 2.0.1
Django	1.9.9	➡ 1.10.5
GitPython	2.0.8	➡ 2.1.1
Jinja2	2.8	➡ 2.8.1
MarkupSafe	0.23	0.23
Metafone	0.5	0.5
PyYAML	3.12	3.12
Pygments	2.1.3	2.1.3

3) Creating your Run/debug configurations

At this step, we will try to run a Lino project under Pycharm. Thus, we need to clone and fetch (Step 1 and 2) the book project . I picked this project because it contains several demo projects. You need also to install it within your virtual python environment.
Once the book project is ready , we choose the ‘min1’ project as a sample.
We create the Run configuration like the following screen-shot




Create Run/Debug Configuration: 'manage'
Configuration
Logs
Before launch: Activate tool window
There are no tasks to run before launch
OK
Cancel
Apply
Help

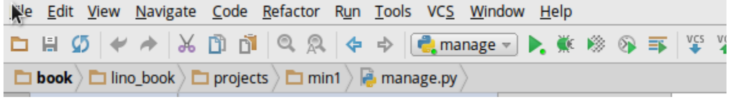
Name: manage   □ Share   □ Single instance only

Script: ts/lino/book/lino_book/projects/min1/manage.py
Script parameters: runserver
Environment
Project: book
Environment variables: PYTHONUNBUFFERED=1
Python interpreter: Project Default (Python 2.7.12 virtualenv at ~/.virtu
Interpreter options:
Working directory: harmProjects/lino/book/lino_book/projects/min1
☑ Add content roots to PYTHONPATH
☑ Add source roots to PYTHONPATH
□ Show command line afterwards

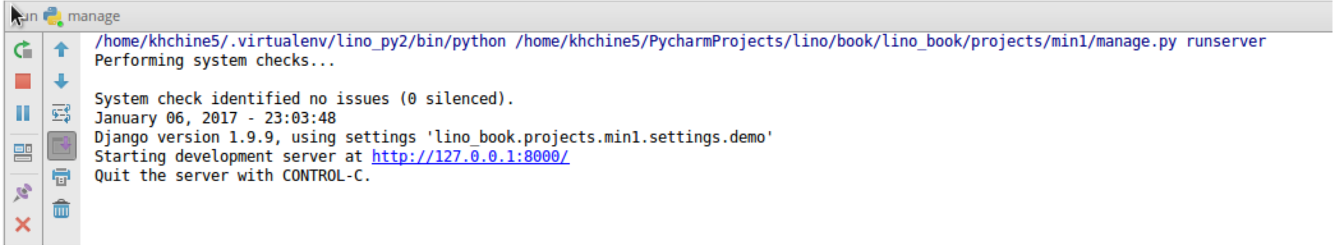
□ Show this page   ☑ Activate tool window

As you can see, this Run configuration will use our configured Python interpreter in Step 2.
4) Run a demo project

Once you create this configuration, you will be able to run your project by clicking on the green arrow (at the right of ‘manage’)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
manage
book
lino_book
projects
min1
manage.py

And you will get a similar output in your console

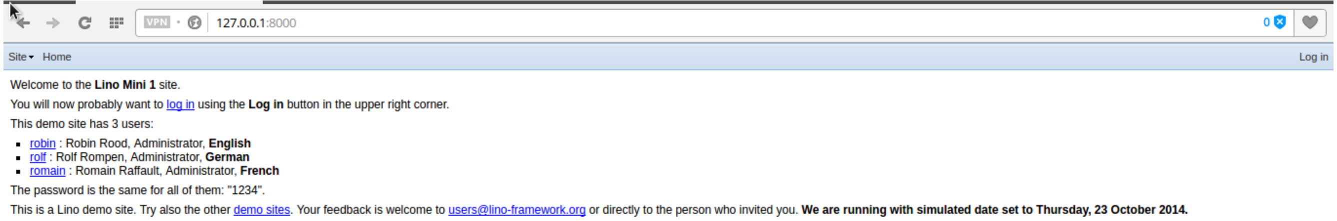


```
/home/khchine5/.virtualenv/lino_py2/bin/python /home/khchine5/PycharmProjects/lino/book/lino_book/projects/min1/manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
January 06, 2017 - 23:03:48
Django version 1.9.9, using settings 'lino_book.projects.min1.settings.demo'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Before opening our browser, we need to populate our database the demo data by going to the book project and run the following command:
‘inv prep’

Now , we can open our browser at this link http://127.0.0.1:8000 and we get our first demo project running.



Welcome to the **Lino Mini 1** site.

You will now probably want to log in using the **Log in** button in the upper right corner.

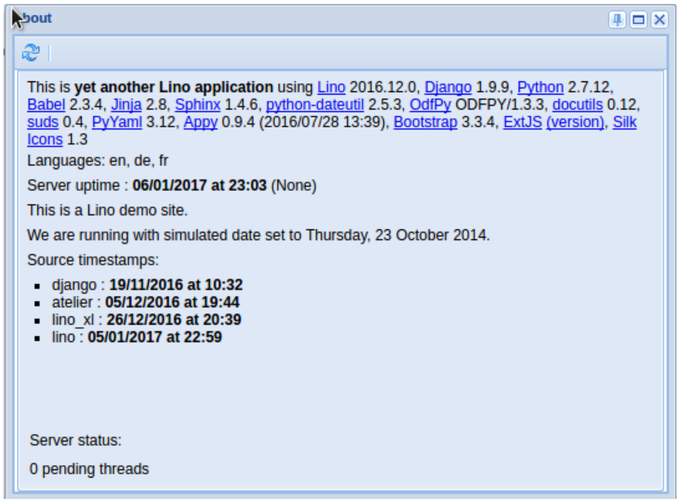
This demo site has 3 users:

- robin : Robin Rood, Administrator, **English**
- rolf : Rolf Rompen, Administrator, **German**
- romain : Romain Raffault, Administrator, **French**

The password is the same for all of them: "1234".

This is a Lino demo site. Try also the other demo sites. Your feedback is welcome to users@lino-framework.org or directly to the person who invited you. **We are running with simulated date set to Thursday, 23 October 2014.**

5) Debug the demo project
Let’s for example debug the “About” window. We can found this window by opening the menu Site>About.



This is **yet another Lino application** using Lino 2016.12.0, Django 1.9.9, Python 2.7.12, Babel 2.3.4, Jinja 2.8, Sphinx 1.4.6, python-dateutil 2.5.3, OdfPy ODFPY/1.3.3, docutils 0.12, suds 0.4, PyYaml 3.12, Appy 0.9.4 (2016/07/28 13:39), Bootstrap 3.3.4, ExtJS (version), Silk Icons 1.3

Languages: en, de, fr

Server uptime : **06/01/2017 at 23:03** (None)

This is a Lino demo site.

We are running with simulated date set to Thursday, 23 October 2014.

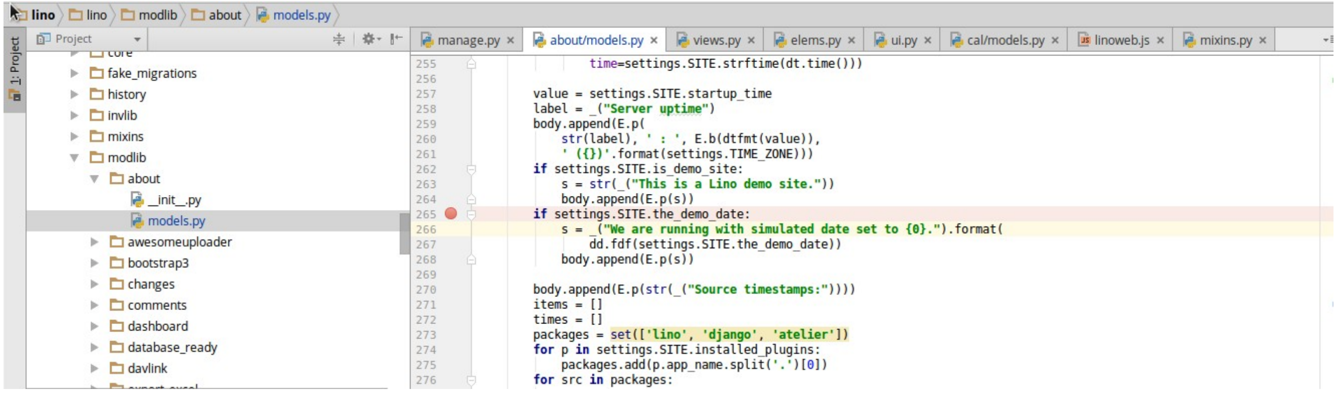
Source timestamps:

- django : **19/11/2016 at 10:32**
- atelier : **05/12/2016 at 19:44**
- lino_xl : **26/12/2016 at 20:39**
- lino : **05/01/2017 at 22:59**

Server status:

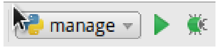
0 pending threads

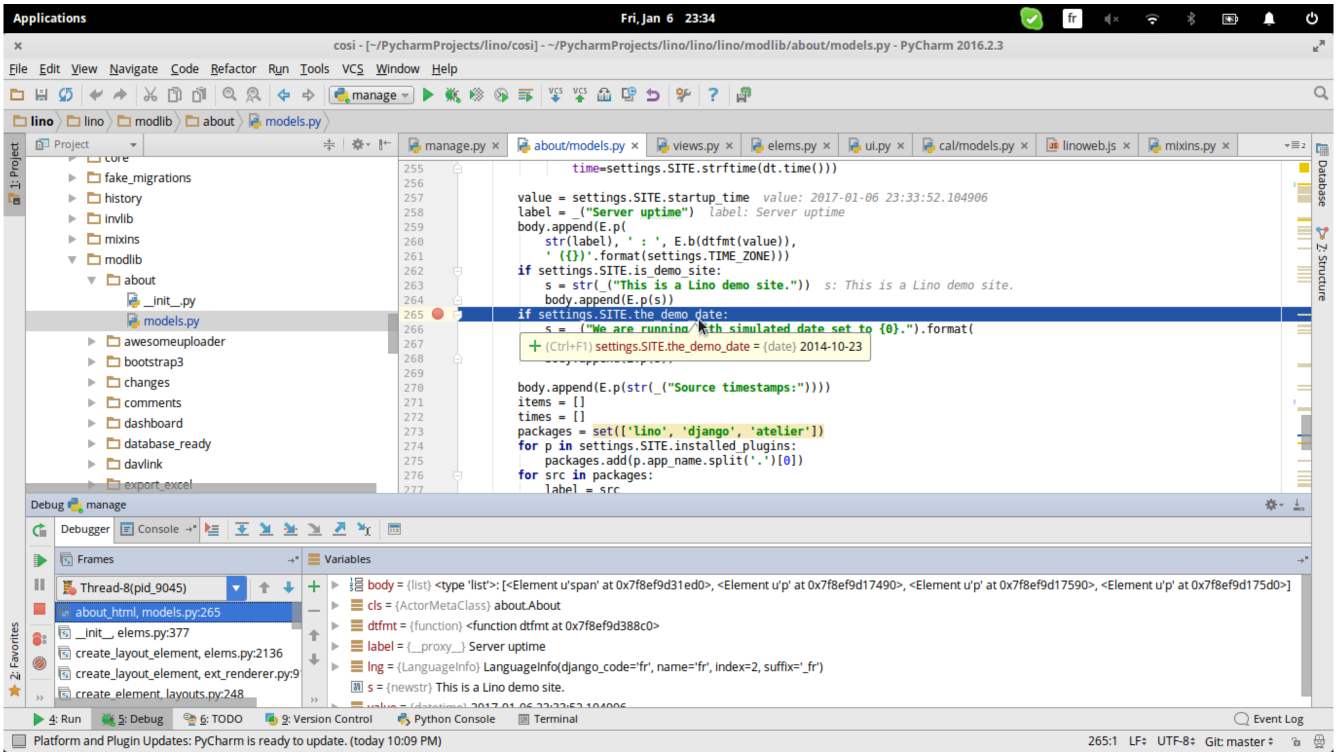
For fun, let’s debug the simulated date in this demo project.
In Pycharm ,we open the file *lino/lino/modlib/about/models.py*



```
255                 time=settings.SITE.strftime(dt.time()))
256 
257         value = settings.SITE.startup_time
258         label = _("Server uptime")
259         body.append(E.p(
260             str(label), ' : ', E.b(dtfmt(value)),
261             ' ({})'.format(settings.TIME_ZONE)))
262         if settings.SITE.is_demo_site:
263             s = str(_("This is a Lino demo site."))
264             body.append(E.p(s))
265         if settings.SITE.the_demo_date:
266             s = _("We are running with simulated date set to {0}.").format(
267                 dd.fdf(settings.SITE.the_demo_date))
268             body.append(E.p(s))
269 
270         body.append(E.p(str(_("Source timestamps:"))))
271         items = []
272         times = []
273         packages = set(['lino', 'django', 'atelier'])
274         for p in settings.SITE.installed_plugins:
275             packages.add(p.app_name.split('.')[0])
276         for src in packages:
```

I added a new breakpoint at the line 265 , I have stopped the run of my application and I started the debug mode by clicking on the button at the right.
I open again my “About” window and the execution stop in my breakpoint





cosi - [~/PycharmProjects/lino/cosi] - ~/PycharmProjects/lino/lino/lino/modlib/about/models.py - PyCharm 2016.2.3

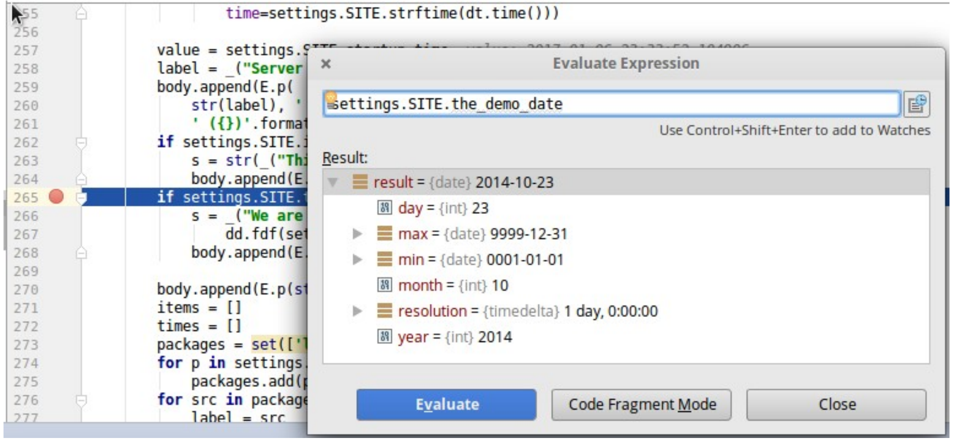
```
255                 time=settings.SITE.strftime(dt.time()))
256 
257         value = settings.SITE.startup_time  value: 2017-01-06 23:33:52.104906
258         label = _("Server uptime")  label: Server uptime
259         body.append(E.p(
260             str(label), ' : ', E.b(dtfmt(value)),
261             ' ({})'.format(settings.TIME_ZONE)))
262         if settings.SITE.is_demo_site:
263             s = str(_("This is a Lino demo site."))  s: This is a Lino demo site.
264             body.append(E.p(s))
265         if settings.SITE.the_demo_date:
266             s = _("We are running with simulated date set to {0}.").format(
267     + (Ctrl+F1) settings.SITE.the_demo_date = {date} 2014-10-23
268 
269 
270         body.append(E.p(str(_("Source timestamps:"))))
271         items = []
272         times = []
273         packages = set(['lino', 'django', 'atelier'])
274         for p in settings.SITE.installed_plugins:
275             packages.add(p.app_name.split('.')[0])
276         for src in packages:
277             label = src
```

```
Thread-8(pid_9045)
about_html, models.py:265
__init__, elems.py:377
create_layout_element, elems.py:2136
create_layout_element, ext_renderer.py:9
create_element, layouts.py:248
```

```
body = {list} <type 'list'>: [<Element u'span' at 0x7f8ef9d31ed0>, <Element u'p' at 0x7f8ef9d17490>, <Element u'p' at 0x7f8ef9d17590>, <Element u'p' at 0x7f8ef9d175d0>]
cls = {ActorMetaClass} about.About
dtfmt = {function} <function dtfmt at 0x7f8ef9d388c0>
label = {__proxy__} Server uptime
lng = {LanguageInfo} LanguageInfo(django_code='fr', name='fr', index=2, suffix='_fr')
s = {newstr} This is a Lino demo site.
value = {datetime} 2017-01-06 23:33:52.104906
```

Platform and Plugin Updates: PyCharm is ready to update. (today 10:09 PM)   265:1 LF÷ UTF-8÷ Git: master ÷

You can use the Evaluate Expression feature by taping Ctrl+U



```
                time=settings.SITE.strftime(dt.time()))

        value = settings.S
        label = _("Server
        body.append(E.p(
            str(label), '
            ' ({})'.format
        if settings.SITE.i
            s = str(_("Thi
            body.append(E.
        if settings.SITE.t
            s = _("We are
                dd.fdf(set
            body.append(E.

        body.append(E.p(st
        items = []
        times = []
        packages = set(['l
        for p in settings.
            packages.add(p
        for src in package
            label = src
```

```
settings.SITE.the_demo_date
```

Result:

```
▼ result = {date} 2014-10-23
      day = {int} 23
  ▶   max = {date} 9999-12-31
  ▶   min = {date} 0001-01-01
      month = {int} 10
  ▶   resolution = {timedelta} 1 day, 0:00:00
      year = {int} 2014
```