
Nobinobi Daily Follow-Up Documentation

Release 0.1.0

Florian Alu

Sep 17, 2020

Contents

1	Nobinobi Daily Follow-Up	3
1.1	Documentation	3
1.2	Quickstart	3
1.3	Features	4
1.4	Running Tests	4
1.5	Development commands	4
1.6	Credits	4
2	Installation	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	History	17
6.1	0.1.0 (2020-09-16)	17

Contents:

Module Daily follow-up for nobinobi

1.1 Documentation

The full documentation is at <https://nobinobi-daily-follow-up.readthedocs.io>.

1.2 Quickstart

Install Nobinobi Daily Follow-Up:

```
pip install nobinobi-daily-follow-up
```

Add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'phonenummer_field',  
    'crispy_forms',  
    'django_extensions',  
    'rest_framework',  
    'rest_framework.authtoken',  
    'rest_framework_datatables',  
    'menu',  
    'bootstrap_modal_forms',  
    'widget_tweaks',  
)
```

(continues on next page)

(continued from previous page)

```
'django_select2',
'bootstrap_datepicker_plus',
'nobinobi_core',
'nobinobi_staff',
'nobinobi_child.apps.NobinobiChildConfig',
'nobinobi_daily_follow_up.apps.NobinobiDailyFollowUpConfig',
...
)
```

Add Nobinobi Daily Follow-Up's URL patterns:

```
from nobinobi_core import urls as nobinobi_core_urls
from nobinobi_staff import urls as nobinobi_staff_urls
from nobinobi_child import urls as nobinobi_child_urls
from nobinobi_daily_follow_up import urls as nobinobi_daily_follow_up_urls

urlpatterns = [
    ...
    path('', include(nobinobi_core_urls)),
    path('', include(nobinobi_staff_urls)),
    path('', include(nobinobi_child_urls)),
    path('', include(nobinobi_daily_follow_up_urls)),
    ...
]
```

1.3 Features

- TODO

1.4 Running Tests

Does the code actually work?

```
source <YOURVIRTUALENV>/bin/activate
(myenv) $ pip install tox
(myenv) $ tox
```

1.5 Development commands

```
pip install -r requirements_dev.txt
invoke -l
```

1.6 Credits

Tools used in rendering this package:

- Cookiecutter

- cookiecutter-djangopackage

CHAPTER 2

Installation

At the command line:

```
$ easy_install nobinobi-daily-follow-up
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv nobinobi-daily-follow-up  
$ pip install nobinobi-daily-follow-up
```


To use Nobinobi Daily Follow-Up in a project, add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (  
    ...  
    'nobinobi_daily_follow_up.apps.NobinobiDailyFollowUpConfig',  
    ...  
)
```

Add Nobinobi Daily Follow-Up's URL patterns:

```
from nobinobi_daily_follow_up import urls as nobinobi_daily_follow_up_urls  
  
urlpatterns = [  
    ...  
    url(r'^$', include(nobinobi_daily_follow_up_urls)),  
    ...  
]
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/prolibre-ch/nobinobi-daily-follow-up/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

Nobinobi Daily Follow-Up could always use more documentation, whether as part of the official Nobinobi Daily Follow-Up docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/prolibre-ch/nobinobi-daily-follow-up/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *nobinobi-daily-follow-up* for local development.

1. Fork the *nobinobi-daily-follow-up* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/nobinobi-daily-follow-up.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv nobinobi-daily-follow-up
$ cd nobinobi-daily-follow-up/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 nobinobi_daily_follow_up tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/prolibre-ch/nobinobi-daily-follow-up/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_nobinobi_daily_follow_up
```


5.1 Development Lead

- Florian Alu <alu@prolibre.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.1.0 (2020-09-16)

- First release on PyPI.