

---

# PyCon APAC 2014 Documentation

*Release 2014-01-12*

**Keith Yang**

July 06, 2014



<b>1</b>	<b>PyCon APAC 2014</b>	<b>3</b>
1.1	Getting Started . . . . .	3
1.2	Setting up the database . . . . .	3
1.3	Running a web server . . . . .	4
1.4	Compile static . . . . .	4
1.5	For Python developers . . . . .	4
1.6	How to contribute . . . . .	5
<b>2</b>	<b>Contributing</b>	<b>7</b>
2.1	Types of Contributions . . . . .	7
2.2	Get Started! . . . . .	8
2.3	Pull Request Guidelines . . . . .	8
2.4	Tips . . . . .	8
<b>3</b>	<b>Credits</b>	<b>11</b>
3.1	Development Lead . . . . .	11
3.2	Chief Developer . . . . .	11
3.3	Chief Design . . . . .	11
3.4	Code Contributors . . . . .	11
3.5	Site content and translators . . . . .	11
<b>4</b>	<b>CHANGELOG</b>	<b>13</b>
4.1	2014-01-12 . . . . .	13
<b>5</b>	<b>LICENSE</b>	<b>15</b>
<b>6</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



---

## PyCon APAC 2014

---

This repository serves the website of PyCon APAC 2014. This project is open source and the license can be found in LICENSE.

Full documentation can be found on [Read the Docs](#).

### 1.1 Getting Started

#### 1.1.1 Requirements

- Git 1.8+
- Python 2.7+

Clone this repository:

```
$ git clone git@bitbucket.org:pycontw/pycon-apac-2014.git
```

#### 1.1.2 Setting up environment

Change to cloned directory path:

```
$ cd pycon-apac-2014
```

Setup its virtualenv and requirements:

```
$ python scripts/make_virtualenv.py .venv
```

Which will create a default virtualenv path ".venv" inside pycon-apac-2014 folder.

Then install the required packages in this virtualenv (.venv):

```
$ .venv/bin/pip install -r requirements.txt
```

### 1.2 Setting up the database

This will vary for production and development. By default the project is set up to run on a SQLite database. For deployment of testing or production:

```
$ .venv/bin/fab deploy
```

If you are setting up a production database see the Configuration section below for where to place settings and get the database running. Now you can run:

```
$ .venv/bin/fab deploy:production
```

## 1.3 Running a web server

In development you should run:

```
$ .venv/bin/fab serve
```

This will run a web server with default port, 8000. If you want to run the server in a specific port and/or host, consider these examples:

```
$ .venv/bin/fab serve:port=8090
$ .venv/bin/fab serve:host=127.0.0.1
$ .venv/bin/fab serve:host=192.168.1.15,port=8090
$ .venv/bin/fab serve:192.168.1.15,8090
```

## 1.4 Compile static

Under conweb/static:

```
$ sass --watch scss/all.scss:all.css
$ coffee --join all.js -cw coffees/*.coffee
```

## 1.5 For Python developers

### 1.5.1 Live in a shell

Want to run `django-admin.py shell`? You can just:

```
$ .venv/bin/fab shell
```

or if you like to go with IPython or bpython:

```
$ .venv/bin/fab shell:interface=ipython
$ .venv/bin/fab shell:interface=bpython
```

You're recommended to use IPython with it. Install IPython by:

```
$ .venv/bin/pip install IPython
```

Note: if you had IPython or bpython installed, `django-admin.py shell` will use it automatically.

### 1.5.2 Question about ".venv/bin/fab"

Can I use `fab . . .` without virtualenv prefix `".venv/bin/"`? It's annoying.



### 1.5.3 Answer

Yes you can. Activate installed virtualenv like `source .venv/bin/activate`. There you can go with `fab` commands.

Or, if you already have fabric installed system-wide, you can just run `fab . . .` without virtualenv prefix path needed.

### 1.5.4 Question about my favor of virtualenv

Can I use my favor of virtualenv version and path? e.g. (`~/virtualenvs/pycontw`)

### 1.5.5 Answer

Yes you can. Create a file named “venv\_prefix” inside `pycon-apac-2014` folder and put (or replace) your own created virtualenv path in first line. The example of file content:

```
~/virtualenvs/pycontw
```

You will need to replace above virtualenv prefix `.venv/bin/*` with `{your_virtualenv_path}/bin/*`; or source your virtualenv before hand.

## 1.6 How to contribute

See `CONTRIBUTING.rst`.

*CONTRIBUTING*



---

## Contributing

---

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

### 2.1 Types of Contributions

#### 2.1.1 Report Bugs

Report bugs at <https://bitbucket.org/pycontw/pycon-apac-2014/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.

#### 2.1.2 Fix Bugs

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

#### 2.1.3 Implement Features

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

#### 2.1.4 Write Documentation

PyCon APAC 2014 could always use more documentation, whether as part of the official PyCon APAC 2014 docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 2.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://bitbucket.org/pycontw/pycon-apac-2014/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 :)

## 2.2 Get Started!

Ready to contribute? Here's how to set up *pycon-apac-2014* for local development.

1. Fork the *pycon-apac-2014* repo on Bitbucket.

2. Clone your fork locally:

```
$ git clone git@bitbucket.org:your_name_here/pycon-apac-2014.git
```

3. Bootstrap your local PyCon APAC 2014 development. See *Setting up environment* in README.rst.

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 pycon-apac-2014 tests
$ py.test tests
$ tox
```

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

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

```
$ 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 Bitbucket website.

## 2.3 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 2.7 and 3.3.

## 2.4 Tips

To run a subset of tests:

```
$ py.test tests
```



### 3.1 Development Lead

- Keith Yang <yang@keitheis.org>

### 3.2 Chief Developer

- Jim Yeh (Chang-Ching) <lemonlatte@gmail.com>

### 3.3 Chief Design

- Art Pai <art@stylecoder.tw>

### 3.4 Code Contributors

- Tzu-ping Chung
- Chia-Chi Chang

### 3.5 Site content and translators

- PyCon Taiwan committees
  - Special thanks to our Chief Editor, Liang Bo Wang <ccwang002@gmail.com>.
  - <https://github.com/pycontw/PyconAPACDraft/graphs/contributors>
  - Chinese: [https://www.transifex.com/projects/p/pycon-apac-2014/language/zh\\_TW/members](https://www.transifex.com/projects/p/pycon-apac-2014/language/zh_TW/members)
  - Japanese: <https://www.transifex.com/projects/p/pycon-apac-2014/language/ja/members>





---

CHANGELOG

---

**4.1 2014-01-12**

- Prepare for *Read the Docs*.
- Use cookiecutter for documentation template.
- Start release with documentation.



---

**LICENSE**

---

The MIT License

Copyright (c) 2013, 2014 PyCon Taiwan. <http://tw.pycon.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*