
Confire Documentation

Release 0.1.0

Benjamin Bengfort

July 23, 2014

1	Features	3
2	Example Usage	5
3	Setup	7
4	About	9
4.1	Contributing	9
4.2	Name Origin	9
5	Changelog	11
5.1	v0.1.1 released on 24 July 2014	11
5.2	v0.1.0 released on 20 July 2014	11
6	Indices and tables	13

Confire is a simple but powerful configuration scheme that builds on the configuration parsers of Scapy, elasticsearch, Django and others. The basic scheme is to have a configuration search path that looks for YAML files in standard locations. The search path is hierarchical (meaning that system configurations are overloaded by user configurations, etc). These YAML files are then added to a default, class-based configuration management scheme that allows for easy development.

Features

- Configuration files in YAML
- Hierarchical configuration search
- Class based application defaults
- Settings pulled in from the environment

Example Usage

Create a file called “myapp.yaml” and place it in one of the following places:

- /etc/myapp.yaml
- \$HOME/.myapp.yaml
- conf/myapp.yaml

Create some configuration values inside the file like so:

```
## Set application environment
debug: True
testing: False

## A simple database configuration
database:
  name: mydb
  host: localhost
  port: 5432
  user: postgres
```

In your code, create a file called “config.py” and add the following:

```
import os
from confire import Configuration

class DatabaseConfiguration(Configuration):

    host = "localhost"
    port = 5432
    name = "mydb"
    user = "postgres"
    password = os.environ.get("DATABASE_PASSWORD", "")

class MyAppConfiguration(Configuration):

    CONF_PATHS = [
        '/etc/myapp.yaml',
        os.path.expanduser('~/.myapp.yaml')
        os.path.abspath('conf/myapp.yaml')
    ]

    debug = False
    testing = True
    database = DatabaseConfiguration()
```

```
settings = MyAppConfiguration.load()
```

Now, everywhere in your code that you would like to access these settings values, simply use as follows:

```
from config import settings
```

```
debug = settings.get('DEBUG') or settings['DEBUG']
```

Voila! A complete configuration system for your application!

Setup

The easiest and usual way to install confire is to use pip:

```
pip install confire
```

To install the package from source, download the latests package tarball, unzip in a temporary directory and run the following command:

```
python setup.py install
```

As always, I highly recommend the use of a virtual environment to better manage the software dependencies for your particular code base.

About

There are many configuration packages available on PyPI - it seems that everyone has a different way of doing it. However, this is my preferred way, and I found that after I copy and pasted this code into more than 3 projects that it was time to add it as a dependency via PyPI. The configuration builds on what I've learned/done in configuring Scapy, elasticsearch, and Django - and builds on these principles:

1. Configuration *should not* be Python (sorry Django). It's too easy to screw stuff up, and anyway, you don't want to deal with importing a settings file from `/etc!`
2. Configuration should be on a per-system basis. This means that there should be an `/etc/app.yaml` configuration file as well as a `$HOME/.app.yaml` configuration file that overwrites the system defaults for a particular user. For development purposes there should also be a `$(pwd)/app.yaml` file so that you don't have to sprinkle things throughout the system if not needed.
3. Developers should be able to have reasonable defaults already written in code if no YAML file has been provided. These defaults should be added in an API like way that is class based and modularized.
4. Accessing settings from the code should be easy.

So there you have it, with these things in mind I wrote confire and I hope you enjoy it!

4.1 Contributing

Confire is open source, and I would be happy to have you contribute! You can contribute in the following ways:

1. Create a pull request in Github: <https://github.com/bbengfort/confire>
2. Add issues or bugs on the bugtracker: <https://github.com/bbengfort/confire/issues>
3. Checkout the current dev board on waffle.io: <https://waffle.io/bbengfort/confire>

You can contact me on Twitter if needed: @bbengfort

4.2 Name Origin

I like cooking, and the thought of preparation in French culinary language appealed to me. The way I got here was to simply change the "g" in config to a "t". A definition lookup and boom, a name!

Changelog

The release versions that are sent to the Python package index are also tagged in Github. You can see the tags through the Github web application and download the tarball of the version you'd like. Additionally PyPI will host the various releases of confire.

The versioning uses a three part version system, "a.b.c" - "a" represents a major release that may not be backwards compatible. "b" is incremented on minor releases that may contain extra features, but are backwards compatible. "c" releases are bugfixes or other micro changes that developers should feel free to immediately update to.

5.1 v0.1.1 released on 24 July 2014

- **tag:** v0.1.1
- **deployment:** July 24, 2014
- **commit:** –

Added Python 3.3 support thanks to [[@tyrannosaurus](https://github.com/tyrannosaurus)](<https://github.com/tyrannosaurus>) who contributed to the changes that would ensure this support for the future. I also added Python 3.3 travis testing and some other minor changes.

5.2 v0.1.0 released on 20 July 2014

- **tag:** v0.1.0
- **deployment:** July 20, 2014
- **commit:** –

Initial deployment of the confire library.

Indices and tables

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