Boss Documentation

Release 0.9.14

BJ Dierkes

July 14, 2016

Contents

The Boss project provides 'Baseline Open Source Software' templates and development tools. It has similarities to Paste Script with regards to templating, but far easier to extend. Though Boss is written in Python, it is not limited to any particular language. Boss is very beta... any feedback is welcome and appreciated. Features:

- Supports multiple local, and remote (git) template sources
- Templates can easily be created and customized to the developers needs
- Support for remote files within templates (i.e. github/gitignore files)

Other sites that might be helpful.

- RTFD: http://boss.readthedocs.org/en/latest/
- CODE: https://github.com/datafolklabs/boss/
- CI: http://travis-ci.org/#!/datafolklabs/boss

Official templates are available from:

- http://github.com/datafolklabs/boss-templates

Contents:

CHAPTER 1

ChangeLog

All bugs/feature details can be found at:

https://github.com/datafolklabs/boss/issues/XXXXX

Where XXXXX is the 'Issue #' referenced below. Additionally, this change log is available online at:

http://boss.rtfd.org/en/latest/changes/

1.1 0.9.14 - Feb 19, 2014

Bugs:

Issue #15 - AttributeError: 'module' object has no attribute 'defaults' Issue #16 - Updates for Cement 2.2

Features:

None

1.2 0.9.12 - Wed Aug 14, 2013

Initial Release. Future releases will mention bugs/features here.

CHAPTER 2

License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Data Folk Labs, LLC nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

CHAPTER 3

Developer Documentation

3.1 Quick Start

The following outlines installation of Boss, as well as quick starting a few applications from the official repository.

3.1.1 Development Environment

It is recommended to work out of a VirtualENV for development of your application. That said, you likely don't want to install boss every time you start a new project, therefore in this case you should consider installing boss to your global system outside of your virtualenv. In most cases you will be creating your project with Boss before creating a virtualenv anyhow.

3.1.2 Installation

Stable versions of Boss are available via PyPi:

```
$ pip install boss
```

To install development versions of Boss you will need to checkout the 'master' branch from GitHub.

```
$ pip install -e git+git://github.com/datafolklabs/boss.git#egg=boss
```

3.1.3 Working with Sources

Boss supports multiple local and remote (git) template repositories. You can see these repositories via the following command:

```
$ boss sources

--         Label: boss
    Source Path: git@github.com:datafolklabs/boss-templates.git
          Cache: /Users/derks/.boss/cache/tmpJDGhlX
     Local Only: False
 Last Sync Time: never
```

You will notice in the above example that the 'boss' repository has never been synced (which will be the case on a new install). To sync templates with remote sources, execute the following:

```
$ boss sync
Syncing Boss Templates . . .
remote: Counting objects: 137, done.
remote: Compressing objects: 100% (73/73), done.
remote: Total 102 (delta 45), reused 83 (delta 26)
Receiving objects: 100% (102/102), 63.38 KiB, done.
Resolving deltas: 100% (45/45), completed with 18 local objects.
From github.com:datafolklabs/boss-templates
   8626879..8bc867a  master     -> origin/master
```

You can add your own sources like so:

```
$ boss add-source my-remote git@github.com:john.doe/boss-templates.git

$ boss add-source local /path/to/my/templates --local
```

The first example is a remote git repository that holds Boss templates. The second example is a local repository only, and will not attempt to sync with a remote upstream repo. At this time, Boss only support remote Git repositories.

3.1.4 Working with Templates

Once your sources are in place, you can see what templates are available to work with:

```
$ boss templates

Local Templates
--------------------------------------------------------------------------------
python
my-custom-template

Boss Templates
--------------------------------------------------------------------------------
cement-script
e2e
license
python
```

To create a new project, or part of a project, from a template do the following:

```
$ boss create ./helloworld -t local:python
Version: [0.9.1]
Python Module Name: helloworld
Python Class Prefix: HelloWorld
Project Name: Hello World
Project Description: Hello World does Amazing Things
Project Creator: [BJ Dierkes]
Project Creator Email: [derks@bjdierkes.com]
License: [BSD-three-clause]
Project URL: http://helloworld.example.com
--------------------------------------------------------------------------------
Writing: /Volumes/Users/derks/helloworld/README
Writing: /Volumes/Users/derks/helloworld/requirements.txt
Writing: /Volumes/Users/derks/helloworld/setup.cfg
Writing: /Volumes/Users/derks/helloworld/setup.py
Writing: /Volumes/Users/derks/helloworld/helloworld/__init__.py
Writing: /Volumes/Users/derks/helloworld/tests/test_helloworld.py
Writing: /Volumes/Users/derks/helloworld/.gitignore
Writing: /Volumes/Users/derks/helloworld/LICENSE
```

You'll notice a few things in this example:

Some questions were pre-populated by default answers. These can be set under an '[answers]' config section in '~/.boss/config'. For example:

```
[answers]
creator = BJ Dierkes
email = derks@bjdierkes.com
version = 0.9.1
license = BSD-three-clause
```

Also, as this is a python project template, the latest 'Python.gitignore' file was pulled down from http://github.com/github/gitignore and copied to .gitignore.

And it works:

```
$ python
>>> import helloworld
```

With tests:

```
$ nosetests
test_helloworld (test_helloworld.HelloWorldTestCase) ... ok

----------------------------------------------------------------------
Ran 1 test in 0.006s

OK
```

3.2 Creating Custom Templates

Boss templates are extremely easy to write. A basic template consists of the following:

- A boss.json config file
- One or more files/directories to copy for new projects

Boss doesn't care what the contents of template files are, as long as it is a readable file. The following is an example python template:

```
-> python
    `----> @module@/
            `----> @module@.py

    `----> setup.py
    `----> setup.cfg
    `----> README
    `----> LICENSE
    `----> boss.yml
```

In this example, '@module@' is a variable defined in the 'boss.yml' config for this template. When calling 'boss create -t local:python' Boss will ask the user to supply a value for '@module@', and then when files/directories are copied that value will be replace at every occurrence of '@module@' both in file/directory names as well as file contents.

3.2.1 Boss Template Configuration Files

Currently Boss only supports a 'boss.json' or boss.yml configuration file which is in the JSON or Yaml format respectively. The following is an example Yaml configuration file:

```
variables:
    version: Version
    module: Python Module Name
    class_prefix: Python Class Prefix
    project: Project Name
    description: Project Description
    creator: Project Creator
    email: Project Creator Email
    url: Project URL
    license: Project License

external_files:
    .gitignore: https://raw.github.com/github/gitignore/master/Python.gitignore
    LICENSE: https://raw.github.com/datafolklabs/license/master/@license@
```

The 'variables' setting is a list of lists. The first item in a list is the variable that is set, and the second is the question as presented to the user for input.

The 'external_files' is also a list of lists, and is optional. These are files that are pulled down externally. The first item in an external file definition is the destination path where that file should be saved to. The second is the remote URL to pull the contents from. In the above example we use external files to pull down a current .gitignore file from Github, as well as a LICENSE file for the given license if it exists.

3.2.2 Working with Variables

Boss treats all variables as strings. Therefore, it supports strings operations during the replacement process. For example, if I had a variable of 'foo', then in my templates I would reference that variable as '@foo@'. If the value of '@foo@' were 'bar' for example, I could do things like:

- @foo.capitalize@ => Bar
- @foo.upper@ => BAR
- @foo.lower@ => bar
- @foo.title@ = Bar

These simple string operations are commonly used throughout templates. That said, don't get carried away ... Boss doesn't intend to be a robust templating language, but rather a facility to easily build and copy templates for new projects.

3.2.3 Delimiter

The default delimiter is '@'. In some cases, this might not work for your template. You can change this in your boss.yml or boss.json config:

```
delimiter: '%'
```

CHAPTER 4

API Reference

FIX ME

CHAPTER 5

Indices and tables

- genindex
- modindex
- search