Come to the Dark Side!
We have a whole bunch of Cookiecutters!

Raphael Pierzina @hackebrot
Raphael Pierzina

• Background in 3D Animation
• Core Developer of Cookiecutter
• Volunteer for Adopt Pytest Month
• Technical Reviewer on a Kivy Book
“What is thy bidding, my master?”

–Darth Vader (The Empire Strikes Back)
Can we rename newbies to puppies?

@gvanrossum I like 'padawan' best!

@hackebrot Hm, I think people will think the bar for a padawan too high. Anyone can be a puppy though.
A command-line utility that creates projects from cookiecutters (project templates). E.g. Python package projects, jQuery plugin projects.

Artwork by Audrey Roy Greenfeld
GitHub

A command-line utility that creates projects from cookiecutters (project templates). E.g. Python package projects, jQuery plugin projects.

1,105 commits  13 branches  17 releases  67 contributors

branch: master

cookiecutter

Add changes of #470 and #471 to changelog in HISTORY.rst

hackebrot authored 7 minutes ago

latest commit c729de4591
Features

- Cross-platform: Windows, Mac, and Linux are officially supported.
- Free software distributed under the terms of the BSD license
- Works with Python 2.7, 3.3, 3.4, and PyPy.
- Unicode support for named platforms and Python versions
- Project templates can be in any programming language or markup format
Technologies

- 100% of templating is done with Jinja2
- Support for Git and Mercurial
- Sphinx Docs hosted on Read the Docs
- Running Tox and Pytest *(at 96% test coverage)*
- CLI and user prompting via Click
Community

The core committer team is @audreyr, @pydanny, @michaeljoseph, @pfmoore, and @hackebrot.

We welcome you and invite you to participate.

Up to date 61 contributors and template authors in total.
Audrey Roy Greenfeld @audreyr - Jul 9
The aftermath of me and @pydanii recording our @django0birthday tribute video.
“After EuroPython 2013, Danny and I both came away from the conference with a renewed spirit and enthusiasm to sprint on open source.”

–Audrey Roy Greenfeld
“I had a lot of ideas for packaging various snippets of my existing code as new Python libraries. But I found that I was doing a lot of repetitive work every time I created a new Python package.”

–Audrey Roy Greenfeld
$ git log --author="Raphael Pierzina" --reverse
You underestimate the power of a good template

Creating a Cookiecutter from Scratch
Kivy - Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps.

http://kivy.org/#home
Create Template Root

$ git clone https://github.com/hackebrot/cookie_dozer.git

$ cd cookiedozer

$ tree -a -L 1
.
  .git
  .gitignore
  LICENSE
  README.md
Python Package

$ mkdir -p {{cookiecutter.repo_name}}/{{cookiecutter.repo_name}}

$ cd $_

$ touch __init__.py
$ touch {{cookiecutter.repo_name}}.py

$ touch {{cookiecutter.app_class_name}}.kv
$ touch main.py
# -*- coding: utf-8 -*-

__author__ = '{{cookiecutter.full_name}}'
__email__ = '{{cookiecutter.email}}'
__version__ = '{{cookiecutter.version}}'
# -*- coding: utf-8 -*-

"""
{{cookiecutter.repo_name}}
============================
The root of :class:`{{cookiecutter.app_class_name}}` is created from the kv file.
"""

import kivy
kivy.require('{{cookiecutter.app_class_name}}')

from kivy.app import App

class {{cookiecutter.app_class_name}}(App):
    
    title = '{app_title}'

    def build(self):
        return self.root
#!/usr/bin/env python
#
# -*- coding: utf-8 -*-

from {{cookiecutter.repo_name}} import {{cookiecutter.app_class_name}}

def main():
    {{cookiecutter.app_class_name}}().run()

if __name__ == '__main__':
    main()
#:kivy {{cookiecutter.kivy_version}}

Button:
    text: '{cookiecutter.app_title}'
cookiecutter.json

$ cd ~

$ touch cookiecutter.json
cookiecutter.json

{
    "full_name": "Raphael Pierzina",
    "email": "raphael@hackebrot.de",
    "app_title": "Hello World",
    "app_class_name": "HelloWorldApp",
    "repo_name": "helloworld",
    "version": "0.1.0",
    "kivy_version": "1.8.0"
}
The ability to destroy a planet is insignificant next to the power of Cookiecutter.

Advanced Usage
Templates in Variables

{
    "full_name": "Raphael Pierzina",
    "email": "raphael@hackebrot.de",
    "app_title": "Hello World",
    "app_class_name": "{{cookiecutter.app_title|replace(' ', '')}}App",
    "repo_name": "{{cookiecutter.app_title|replace(' ', '')|lower}}",
    "version": "0.1.0",
    "kivy_version": "1.8.0"
}
Hooks

$ mkdir hooks

$ touch $/post_gen_project.py
Rename the generated kv file to be compatible with the original kivy kv file detection of `App.load_kv()`.

import os

package_dir = '{cookiecutter.repo_name}'
old_kv_file = os.path.join(package_dir, '{cookiecutter.app_class_name}.kv')

lower_app_class_name = '{cookiecutter.app_class_name}'.lower()
if (lower_app_class_name.endswith('app')):
    lower_app_class_name = lower_app_class_name[:-3]
new_kv_file = os.path.join(package_dir, '{}.kv'.format(lower_app_class_name))
os.rename(old_kv_file, new_kv_file)
$ tree

.  cookiecutter.json
  {cookiecutter.repo_name}/
    {cookiecutter.repo_name}/
      {cookiecutter.app_class_name}.kv
      {cookiecutter.repo_name}.py
      __init__.py
      main.py
    LICENSE
    README.rst
  hooks
    post_gen_project.py
    LICENSE
    README.rst
Test Suite

```bash
$ mkdir \{{cookiecutter.repo_name}\}/tests
$ cd $_

$ touch conftest.py
$ touch test_{{cookiecutter.repo_name}}.py
```
import pytest

@pytest.fixture(scope="session")
def app(request):
    """Uses the InteractiveLauncher to provide access to an app instance. The finalizer stops the launcher once the tests are finished.

    Returns:
    :class:`{{cookiecutter.app_class_name}}`: App instance
    """
    from kivy.interactive import InteractiveLauncher
    from {{cookiecutter.repo_name}}.{{cookiecutter.repo_name}} import {{cookiecutter.app_class_name}}

    return InteractiveLauncher(app_class_name="{{cookiecutter.app_class_name}}")
conftest.py

```python
launcher = InteractiveLauncher({{cookiecutter.app_class_name}}())

def stop_launcher():
    launcher.safeOut()
    launcher.stop()

request.addfinalizer(stop_launcher)

launcher.run()
launcher.safeIn()
return launcher.app
```
# -*- coding: utf-8 -*-

def test_app_title(app):
    
    """Simply tests if the default app title meets the expectations.
    
    Args:
    app (:class:`{{cookiecutter.app_class_name}}`): Default app instance
    
    Raises:
    AssertionError: If the title does not match
    """

    assert app.title == '{{cookiecutter.app_title}}'
$ touch {{cookiecutter.repo_name}}/setup.py
Resources on Python Packaging


http://www.jeffknupp.com/blog/2013/08/16/open-sourcing-a-python-project-the-right-way/

http://blog.ionelmc.ro/2014/05/25/python-packaging/

...
Sphinx Documentation

$ cd -

$ mkdir {{cookiecutter.repo_name}}/docs
$ cd $_

$ sphinx-quickstart
Sphinx Quickstart

> Root path for the documentation [.]: <ENTER>
> Separate source and build directories (y/n) [n]: y
> Name prefix for templates and static dir [_]: <ENTER>

> Project name: {{cookiecutter.app_title}}
> Author name(s): {{cookiecutter.full_name}}
> Project version: {{cookiecutter.version}}
> Project release [{cookiecutter.version}]: <ENTER>

> Source file suffix [.rst]: <ENTER>
User Config

$ touch ~/.cookiecutterrc
default_context:
  full_name: "Raphael Pierzina"
  email: "raphael@hackebrot.de"
  github_username: "hackebrot"
abbreviations:
  cookiedozer: https://github.com/hackebrot/cookiedozer.git
  gh: https://github.com/{0}.git
Showcase User Config

$ cookiecutter gh:audreyr/cookiecutter-pypackage

Cloning into 'cookiecutter-pypackage'...
remote: Counting objects: 505, done.
remote: Total 505 (delta 0), reused 0 (delta 0), pack-reused 505
Receiving objects: 100% (505/505), 77.43 KiB | 0 bytes/s, done.
Resolving deltas: 100% (265/265), done.
Checking connectivity... done.
full_name [Raphael Pierzina]:
email [raphael@hackebrot.de]:
github_username [hackebrot]:
project_name [Python Boilerplate]: Test User Config
repo_name [boilerplate]:
project_short_description [Python Boilerplate contains all the boilerplate you need to create a Python package.]:
release_date [2015-01-11]:
year [2015]:
version [0.1.0]:
“Bring my shuttle.”

–Darth Vader (The Empire Strikes Back)
Choices in Context

```json
{
    "full_name": "Raphael Pierzina",
    "email": "raphael@hackebrot.de",
    "github_username": "hackebrot",
    "project_name": "Kivy Project",
    "minimum_kivy_version": "1.8.0",
    "orientation": ["all", "landscape", "portrait"]
}
```
Choice Prompt

Select orientation:
1 - all
2 - landscape
3 - portrait
Choose from 1, 2, 3 [1]: 2
def read_user_choice(var_name, options):
    """Prompt the user to choose from several options for the given variable.

    The first item will be returned if no input happens.

    :param str var_name: Variable as specified in the context
    :param list options: Sequence of options that are available to select from
    :return: Exactly one item of ``options`` that has been chosen by the user
    """

    # Please see http://click.pocoo.org/4/api/#click.prompt
    if not isinstance(options, list):
        raise TypeError

    if not options:
        raise ValueError
Click Prompts

```python
choice_map = OrderedDict(
    ('{}'.format(i), value) for i, value in enumerate(options, 1)
)
choices = choice_map.keys()
default = '1'

choice_lines = ['{} - {}'.format(*c) for c in choice_map.items()]
prompt = '
'.join(
    'Select {}:
    ' + '
'.join(choice_lines),
    'Choose from {}
').format(var_name),

user_choice = click.prompt(
    prompt, type=click.Choice(choices), default=default
)
return choice_map[user_choice]
```
Join us, and together we can rule the galaxy as Cookiecutterers.

Template Contributions
Templates

41 Templates

- Python
- C++
- Common Lisp
- LaTeX/XeTeX
- HTML
- 6502 Assembly
- C
- C#
- JS
- Berkshelf-Vagrant
- Scala
A cookiecutter template for creating Django projects quickly.
Features

- For Django 1.8
- Twitter Bootstrap 3
- AngularJS
- 12-Factor based settings via django-environ
- Optimized development and production settings
- Registration via django-allauth
- Procfiile for deploying to Heroku
- Grunt build for compass and livereload
- Basic e-mail configurations for send emails via Mailgun
- Media storage using Amazon S3
- Serve static files from Amazon S3 or Whitenoise (optional)
cookiecutter-pylibrary
Features

This is an "all inclusive" sort of template.

- BSD 2-clause license.
- **Tox** for managing test environments for Python 2.6, 2.7, 3.3, PyPy etc.
- **Pytest** or **Nose** for testing Python 2.6, 2.7, 3.3, PyPy etc.
- **Optional** support for creating a tests matrix out of dependencies and python versions.
- **Travis-CI** and **AppVeyor** for continuous testing.
- **Coveralls** or **Codecov** for coverage tracking (using **Tox**).
- Documentation with **Sphinx**, ready for **ReadTheDocs**.
- Configurations for:
  - **isort**
  - **bumpversion**
- Support for C extensions (including coverage measurement for the C code).
- Packaging and code quality checks. This template comes with a tox environment (**check**) that will:
  - Check if your **README.rst** is valid.
  - Check if the **MANIFEST.in** has any issues.
  - Run **flake8** (a combo of PEP8, pyflakes and McCabe checks)
Wrapping up Cookiedozer
February 2, 2015 By Raphael in Python
We are about to finish this tutorial on how to create a Cookiecutter template for an Android application running Kivy. If you haven't checked out Create Your Own Cookiecutter and Extending Cookiedozer yet, please have a brief look into these articles as this post draws on...

Extending Cookiedozer
January 18, 2015 By Raphael in Python
Last time we learned how to create a basic Kivy application that will be generated with Cookiecutter. The next logical steps are to add tests and docs. Additionally we will create a setup.py as well as a MAKEFILE. The template it's current state contains the following...

Create Your Own Cookiecutter
January 7, 2015 By Raphael in Python
Are you doing the same steps over and over again every time you start a new programming project? More particularly when creating a PyPi compliant application which features a test suite along with auto-generated documentation? If so, you should definitely check out Cookiecutter....
Personal Blog
www.hackebrot.de

Cookiecutter 3D Artworks by
Anselm Wagner
www.anselmwagner.com
Thank you!