
blueberry.py Documentation

Release latest

July 21, 2015

1	What is it?	3
2	What does it offer?	5
3	Installation	7
4	Usage	9

Same as CherryPy, just with a different filling.

Read the latest documents at <http://blueberry.readthedocs.org/>.

What is it?

BlueberryPy is a CherryPy project skeleton generator and a collection of tools, plugins and utilities for [CherryPy](#) - the minimalist Python web framework.

What does it offer?

BlueberryPy is offered as a set of CherryPy tools, WSPB plugins and extra utility modules. All the components are optional and completely pluggable without any intermodule dependencies, so you can safely pick and choose just the components you want.

- SQLAlchemy ORM plugin with two-phase commit support
- Per-request SQLAlchemy ORM session tool
- Redis session storage
- Jinja2 template engine
- Webassets asset pipeline integrated with Jinja2
- Application specific logging
- CherryPy project skeleton generator
- Preconfigured console for experimenting inside a generated project
- YAML configuration for CherryPy, Python's logging module and Webassets
- Convenient Email module for sending text and HTML emails
- JSON tools to convert to and from SQLAlchemy models
- CSRF token

Installation

Stable version:

```
$ pip install blueberry.py
```

Development version:

```
$ pip install https://bitbucket.org/wyuenho/blueberry.py
```

Note: You should probably install it inside a [virtualenv](#).

Usage

Once BlueberryPy is installed, a script called *blueberry* should be available on your PATH.

```
usage: blueberry [-h] [-v] [-C CONFIG_DIR] [command]

BlueberryPy lightweight pluggable Web application framework command line interface.

Type 'blueberry -h' or 'blueberry --help' for general help.
Type 'blueberry help <command>' for help on that specific command.

commands:

  help          print this help or a command's if an argument is given
  create        create a project skeleton
  console       BlueberryPy REPL for experimentations
  bundle        bundles up web assets (type 'blueberry help bundle' for details)
  serve        spawn a new CherryPy server process

positional arguments:
  command          the action to perform

optional arguments:
  -h, --help      show this help message and exit
  -v, --version   print version information and exit.
  -C CONFIG_DIR, --config_dir CONFIG_DIR
                  path to the config directory
```

To create a project skeleton:

```
$ blueberry create
```

After you've answered a couple of questions, you should see something similar to this:

```
=====
Your project skeleton has been created under /path/to/your/project .

Subsystems chosen
-----
Routes (RESTful controllers): True
Jinja2: True
webassets: True
redis: False
SQLAlchemy: True
...
=====
```

If you install a development version of your package now, the dependencies will be automatically installed for you as well:

```
$ pip install -e .
```

You can install the optional speedup packages too:

```
$ pip install blueberry[speedups]
```

Finally, you need to install a database driver such as *psycpg2*:

```
$ pip install psycpg2
```

Now your package is ready to be served:

```
$ blueberry serve
```

Type `http://localhost:8080` into your browser's location bar and voila!

Happy coding!