
datacoco-core

Release 0.1.0

Mar 11, 2020

Contents

1	Installation	3
2	Quickstart	5
2.1	Logger	5
2.2	config	5
3	Development	7
3.1	Getting Started	7
3.2	Testing	7
3.3	Contributing	7

Equinox Common Code Utility for Python 3 with minimal dependencies and easy installation!

Includes utilities for logging and config files

CHAPTER 1

Installation

datacoco-core requires Python 3.6+

```
python3 -m venv venv
source venv/bin/activate
python -m pip install datacoco_core
```


2.1 Logger

The logging module is a lightweight wrapper around the default logging module. Basic usage:

```
from datacoco_core import Logger
l = Logger()
l.l("Your important log message here")
```

By default the log message will be saved in a logs dir of the project root, in a file named by python module and date. For example:

```
cat logs/test.py-20190827-135736.log
2019-08-27 13:57:36,471 Your important message here
```

Optionally, parameters `logname` and `project_name` can be passed on class instantiation to customize the logfile name, and logfile path respectively.

2.2 config

The config module is a lightweight wrapper around the configparser module. It converts INI files to a dictionary object. By default, the Config class will look for a file named in `etl.cfg` in the project root.

Example INI config file:

```
[secret1]
answer_to_the_universe=42
```

This credential can be accessed by the following example code:

```
from datacoco_core.config import config
c = config()
c['secret1']['answer_to_the_universe']
42
```

Note: The config class assumes base64 hashing for any key named pwd or password. (See below)

Password Encryption

The coco config class uses base64 encryption. Any option named pwd or password will be assumed base 64 encrypted. To derive the encrypted password for your config, launch python shell and run the following command:

```
>>> import base64
>>> print base64.b64encode("password")
cGFzc3dvcmQ=
```

In python3 if you get the error `TypeError: a bytes-like object is required, not 'str'` do this.

```
>>> import base64
>>> print (base64.b64encode(b'password'))
b'cGFzc3dvcmQ='
```

3.1 Getting Started

It is recommended to use the steps below to set up a virtual environment for development:

```
python3 -m venv <virtual env name>
source venv/bin/activate
pip install --upgrade pip
pip install datacoco-core
```

3.2 Testing

```
pip install -r requirements-dev.txt
```

To run the testing suite, simply run the command: `tox` or `python -m unittest discover tests`

3.3 Contributing

Contributions to `datacoco_core` are welcome!

Please reference guidelines to help with setting up your development environment [here](#).