b3j0f.annotation Documentation

Release 0.3.6

b3j0f

September 21, 2016

Table of Contents

CHAPTER 1

Changelog

1.1 0.3.6 (2016/09/21)

- change depth parameter to mindepth and maxdepth.

1.2 0.3.5 (2016/09/20)

- add depth parameter in searching deeply annotations.

1.3 0.3.4 (2016/06/06)

- change type of Annotation.targets from set to list.

1.4 0.3.3 (2016/03/12)

- fix a bug in adding the parameter `ctx` in the call to the super method _bind_target of the Interceptor.
- add the b3j0f.annotation.call.Memoize annontation which is given in order to save function result related to parameters.

1.5 0.3.2 (2016/02/22)

- remove dependency to future.
- add support of cpython.

1.6 0.3.1 (2015/11/09)

- add support for python3.5.

1.7 0.3.0 (2015/11/09)

- add dependency to six and future.

1.8 0.2.3 (2015/09/22)

- add reference to module members in the main package.

1.9 0.2.2 (2015/06/14)

- add the folder docs in order to be hosted by readthedocs.

1.10 0.2.1 (2015/06/02)

- use shields.io badges in the README.

1.11 0.2.0 (2015/06/02)

- fix errors in wheel packaging distribution.

1.12 0.1.6 (2015/05/20)

- add wheel package.

1.13 0.1.5 (2015/02/27)

- Fix bug when trying to annotate a class constructor in python3+.

1.14 0.1.4 (2015/02/27)

- Fix bug when trying to annotate a class constructor.

1.15 0.1.3 (2015/02/14)

- Add selection function in Annotation selection functions.

1.16 0.1.2 (2015/12/02)

- Move code from the package annotation to the module core.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

CHAPTER 3

Description

Annotation library like Java's annotation with reflective concerns for Python.

CHAPTER 4

Links

- Homepage
- PyPI
- Documentation

CHAPTER 5

Installation

pip install b3j0f.annotation

CHAPTER 6

Features

What does mean annotations in a reflective way:

- one annotation can annotate several objects at a time (modules, classes, functions, instances, builtins, annotation like themselves, etc.).
- such as a reflective object, they could have their own behavior and lifecycle independently to annotated elements.

This library provides the base Annotation class in order to specialize your own annotations, and several examples of useful annotation given in different modules such as:

- async: dedicated to asynchronous programming.
- interception: annotations able to intercept callable object calls.
- call: inherits from interception module and provides annotations which allow to do checking on callable objects.
- check: annotations which check some conditions such as type of annotated targets, max number of annotated elements, etc.
- oop: useful in object oriented programming like allowing to weave mixins.

CHAPTER 7

Examples

CHAPTER 8

Perspectives

- Cython implementation.

CHAPTER 9

Donation