
doppel-cli Documentation

Release 0.3.1

James Lamb

Apr 30, 2023

CONTENTS

1	API Documentation	1
1.1	Main Interface	1
2	Indices and tables	5
	Python Module Index	7
	Index	9

API DOCUMENTATION

1.1 Main Interface

imports for doppel. Use `__all__` to add things to the API Reference in the docs in `docs/`.

class `doppel.PackageAPI(pkg_dict)`

Package API class

This class is used to hold the interface of a given package being analyzed by doppel. It's comparison operators enable comparison between interfaces and its standard JSON format allows this comparison to happen across programming languages.

class_names()

Get a list with the names of all exported classes in the package.

Return type

`List[str]`

classmethod `from_json(filename)`

Instantiate a Package object from a file.

Parameters

filename (`str`) – Name of the JSON file that contains the description of the target package's API.

Return type

`PackageAPI`

function_names()

Get a list with the names of all exported functions in the package.

Return type

`List[str]`

functions_with_args()

Get a dictionary with all exported functions in the package and some details describing them.

Return type

`Dict[str, Dict[str, Any]]`

name()

Get the name of the package.

Return type

`str`

num_classes()

Get the number of exported classes in the package.

Return type

`int`

num_functions()

Get the number of exported functions in the package.

Return type

`int`

public_method_args(*class_name*, *method_name*)

Get a list of arguments for a public method from a class.

Parameters

- **class_name** (`str`) – Name of a class in the package
- **method-name** – Name of the method to get arguments for

Return type

`List[str]`

public_methods(*class_name*)

Get a list with the names of all public methods for a class.

Parameters

class_name (`str`) – Name of a class in the package

Return type

`List[str]`

class `doppel.PackageCollection(packages)`

Create a collection of multiple PackageAPI objects. This class contains access methods so you don't have to keep doing for `package` in `packages` over a collection of PackageAPI instances.

all_classes()

List of all classes that exist in at least one of the packages.

Return type

`List[str]`

all_functions()

List of all functions that exist in at least one of the packages.

Return type

`List[str]`

non_shared_classes()

List of all classes that are present in at least one but not ALL packages

Return type

`List[str]`

non_shared_functions()

List of all functions that are present in at least one but not ALL packages

Return type

`List[str]`

package_names()

Get a list of all the package names in this collection.

Return type

`List[str]`

shared_classes()

List of shared classes across all the packages in the collection

Return type

`List[str]`

shared_functions()

List of shared functions across all the packages in the collection

Return type

`List[str]`

shared_methods_by_class()

List of public methods in each shared class across all packages

Return type

`Dict[str, List[str]]`

INDICES AND TABLES

- genindex
- search

PYTHON MODULE INDEX

d

doppel, 1

INDEX

A

`all_classes()` (*doppel.PackageCollection* method), 2
`all_functions()` (*doppel.PackageCollection* method),
2

C

`class_names()` (*doppel.PackageAPI* method), 1

D

`doppel`
module, 1

F

`from_json()` (*doppel.PackageAPI* class method), 1
`function_names()` (*doppel.PackageAPI* method), 1
`functions_with_args()` (*doppel.PackageAPI*
method), 1

M

module
doppel, 1

N

`name()` (*doppel.PackageAPI* method), 1
`non_shared_classes()` (*doppel.PackageCollection*
method), 2
`non_shared_functions()` (*doppel.PackageCollection*
method), 2
`num_classes()` (*doppel.PackageAPI* method), 1
`num_functions()` (*doppel.PackageAPI* method), 2

P

`package_names()` (*doppel.PackageCollection* method),
2
`PackageAPI` (class in *doppel*), 1
`PackageCollection` (class in *doppel*), 2
`public_method_args()` (*doppel.PackageAPI* method),
2
`public_methods()` (*doppel.PackageAPI* method), 2

S

`shared_classes()` (*doppel.PackageCollection*
method), 3
`shared_functions()` (*doppel.PackageCollection*
method), 3
`shared_methods_by_class()` (*doppel.PackageCollection*
method), 3