
kaptan Documentation

Release 0.6.0

Emre Yilmaz

Aug 27, 2023

Contents

1	installation	3
2	usage	5
3	exporting configuration	7
4	cli	9
5	running tests	11
6	contributors	13
6.1	API Reference	13
6.2	Command Line Interface	15
6.3	History	15
6.4	Roadmap	17
	Python Module Index	19
	Index	21

configuration parser.

CHAPTER 1

installation

```
$ pip install kaptan
```

Also available as a package on FreeBSD, Debian, Arch Linux and Slackware.

supported handlers

- dict
- json
- yaml
- .ini
- python file

default (dict) handler

```
config = kaptan.Kaptan()
config.import_config({
    'environment': 'DEV',
    'redis_uri': 'redis://localhost:6379/0',
    'debug': False,
    'pagination': {
        'per_page': 10,
        'limit': 20,
    }
})

print config.get("pagination.limit")

# output: 20
```

json handler

```
config = kaptan.Kaptan(handler="json")
config.import_config('{"everything": 42}')

print config.get("everything")

# output: 42
```

yaml handler

```
config = kaptan.Kaptan(handler="yaml")
config.import_config("""
product:
  price:
    value: 12.65
    currency_list:
      1. TL
      2. EURO
""")
print config.get("product.price.currency_list.0")
# output: TL
```

or you can get from directly from the filename:

```
config.import_config("configuration.yaml")
```

.ini handler

config.ini

```
[development]
database_uri = mysql://root:123456@localhost/posts

[production]
database_uri = mysql://poor_user:poor_password@localhost/poor_posts
```

```
config = kaptan.Kaptan(handler="ini")
config.import_config('config.ini')

print config.get("production.database_uri")
# output: mysql://poor_user:poor_password@localhost/poor_posts
```

file handler

config.py

```
DATABASE = 'mysql://root:123456@localhost/posts'
DEBUG = False
PAGINATION = {
    'per_page': 10,
    'limit': 20,
}
```

```
config = kaptan.Kaptan(handler="file")
config.import_config('config')

print config.get("DEBUG")
# output: False
```

CHAPTER 3

exporting configuration

```
config = kaptan.Kaptan(handler="file")
config.import_config({
    'environment': 'DEV',
    'redis_uri': 'redis://localhost:6379/0',
    'debug': False,
    'pagination': {
        'per_page': 10,
        'limit': 20,
    }
})

print config.export("yaml")
```

output:

```
debug: false
environment: DEV
pagination: {limit: 20, per_page: 10}
redis_uri: redis://localhost:6379/0
```

```
print config.export("json")
```

outputs unindented json. `.export` accepts kwargs which pass into *json.dumps*.

```
print config.export("json", indent=4)
```

output:

```
{
  "environment": "DEV",
  "debug": false,
  "pagination": {
    "per_page": 10,
    "limit": 20
  }
}
```

(continues on next page)

(continued from previous page)

```
    },  
    "redis_uri": "redis://localhost:6379/0"  
}
```

`config.export('yaml')` also supports the [kwargs](#) for `pyyaml`.

New in Version 0.5.7: `config.export('yaml', safe=True)` will use `.safe_dump`.

CHAPTER 4

cli

exporting (defaults to json)

```
$ echo "environment: DEV" > config.yaml
$ kaptan config.yaml --export json > config.json
$ cat config.json
{"environment": "DEV"}
```

getting a value

```
$ kaptan config.yaml --key environment
DEV
```

specifying the handler

```
$ mv config.yaml config.settings
$ kaptan config.settings:yaml --export json
{"environment": "DEV"}
```

config from stdin

```
$ echo '{"source": "stdin"}' | kaptan -
{"source": "stdin"}
$ echo 'source: stdin' | kaptan -:yaml
{"source": "stdin"}
```

merging configs

```
$ echo "environment: PROD" > config.settings
$ echo '{"source": "stdin"}' | kaptan - config.json config.settings:yaml
{"environment": "PROD", "source": "stdin"}
```

setting default handler

```
$ echo "source: stdin" | kaptan --handler yaml - config.settings
{"environment": "PROD", "source": "stdin"}
```

writing json with yaml

```
$ kaptan -:yaml -e json  
<type yaml here>  
<Ctrl + D>  
<get json here>
```

CHAPTER 5

running tests

with `py.test`:

```
$ py.test
```


- Cenk Altı
- Wesley Bitter
- Mark Steve
- Tony Narlock
- Berker Peksag
- Pradyun S. Gedam

see more at <https://github.com/emre/kaptan/graphs/contributors>.

Explore:

6.1 API Reference

6.1.1 kaptan

configuration parser.

copyright

(c) 2013 by the authors and contributors (See AUTHORS file).

license BSD, see LICENSE for more details.

class `kaptan.Kaptan` (*handler=None*)

Bases: `object`

HANDLER_MAP = {'dict': <class 'kaptan.handlers.dict_handler.DictHandler'>, 'file': <

upsert (*key, value*)

_is_python_file (*value*)

Return True if the *value* is the path to an existing file with a *.py* extension. False otherwise

```
import_config (value)
_get (key)
get (key=None, default=<object object>)
export (handler=None, **kwargs)
_Kaptan__handle_default_value (key, default)

kaptan.get_parser ()
    Create and return argument parser.

    Return type argparse.ArgumentParser
    Returns CLI Parser

kaptan.main ()

class kaptan.Kaptan (handler=None)
    Bases: object

    _is_python_file (value)
        Return True if the value is the path to an existing file with a .py extension. False otherwise
```

6.1.2 Handlers

```
class kaptan.handlers.BaseHandler
    Bases: object

    Base class for data handlers.

    dump (data)
    load (data)

class kaptan.handlers.dict_handler.DictHandler
    Bases: kaptan.handlers.BaseHandler

    dump (data)
    load (data)

class kaptan.handlers.ini_handler.IniHandler
    Bases: kaptan.handlers.BaseHandler

    dump (data, file_=None)
    load (value)

class kaptan.handlers.json_handler.JsonHandler
    Bases: kaptan.handlers.BaseHandler

    dump (data, **kwargs)
    load (data)

class kaptan.handlers.yaml_handler.YamlHandler
    Bases: kaptan.handlers.BaseHandler

    dump (data, safe=True, **kwargs)
    load (data, safe=True)
```

6.2 Command Line Interface

Configuration manager in your pocket

```
usage: kaptan [-h] [--handler HANDLER] [-e EXPORT] [-k KEY]
            [config_file [config_file ...]]
```

6.2.1 Positional Arguments

config_file file/s to load config from

6.2.2 Named Arguments

--handler set default handler
 Default: “json”

-e, --export set format to export to
 Default: “json”

-k, --key set config key to get value of

6.3 History

Here you can find the recent changes to kaptan

6.3.1 v0.6.0 (2023-08-27)

Features

- .ini export (#160) thank you @onurguzel

CI

- Add GitHub action CI workflow with *pipenv* and python-version test grid (#251)
- Remove .travis.yml (#251)

Packaging updates

- Remove upper bound on pyyaml version (#252)
 Development commits also bump pyyaml version to from <6 to <7 (#250), thank you @pauloacmartinez
- Remove version pinning from dev packages (#250), thank you @pauloacmartinez
- Update classifiers to currently supported Python versions (#250), thank you @pauloacmartinez
- Fix classifiers warning (#226) by @sunpoet
- Relax pytest version constraint (#163), thank you @ignatenkobrain

Meta

- Fix license language to official BSD license (#234)

6.3.2 v0.5.12 (2019-04-22)

- Bump pipenv version to python 3.x
- Utility package updates
- Bump pyyaml version to <6

6.3.3 v0.5.11 (2018-12-30)

- Update dependencies for pytest, sphinx, etc.
- Forward compatibility with Python 3.8 `collections.abc`
- sphinx-argparse 0.2.2 to 0.2.5
- Sphinx 1.7.5 to 1.8.3
- sphinx-rtd-theme 0.4.0 to 0.4.2
- flake8 3.5.0 to 3.6.0
- pytest 3.6.2 to 4.0.2

6.3.4 v0.5.10 (2018-07-06)

- Update Pipfile
- Relax pyyaml requirements to <4
- Support for Python 3.7
- Prevent test from creating stray .pyc files
- Update pytest 3.2.3 to 3.6.3
- Update sphinx 1.6.4 to 1.7.5
- Update sphinx-argparse 0.2.1 to 0.2.2
- Update sphinx-rtd-theme 0.2.4 to 0.4.0
- Add `make sync_pipfile` for updating Pipfile

6.3.5 v0.5.9 (2017-10-19)

- Make yaml safe by default (<https://github.com/emre/kaptan/pull/46>) by @pradyunsg
- Add `Pipfile` for `pipenv`
- Add `.tmuxp.yaml` for `tmuxp`
- Add `Makefile` with tasks to re-run commands on file save, rebuild sphinx
- Update `py.test`, `sphinx` versions

6.4 Roadmap

See <https://github.com/emre/kaptan/issues>.

k

kaptan, [17](#)

Symbols

`_Kaptan__handle_default_value()` (*kaptan.Kaptan method*), 14
`_get()` (*kaptan.Kaptan method*), 14
`_is_python_file()` (*kaptan.Kaptan method*), 13, 14

B

`BaseHandler` (*class in kaptan.handlers*), 14

D

`DictHandler` (*class in kaptan.handlers.dict_handler*), 14
`dump()` (*kaptan.handlers.BaseHandler method*), 14
`dump()` (*kaptan.handlers.dict_handler.DictHandler method*), 14
`dump()` (*kaptan.handlers.ini_handler.IniHandler method*), 14
`dump()` (*kaptan.handlers.json_handler.JsonHandler method*), 14
`dump()` (*kaptan.handlers.yaml_handler.YamlHandler method*), 14

E

`export()` (*kaptan.Kaptan method*), 14

G

`get()` (*kaptan.Kaptan method*), 14
`get_parser()` (*in module kaptan*), 14

H

`HANDLER_MAP` (*kaptan.Kaptan attribute*), 13

I

`import_config()` (*kaptan.Kaptan method*), 13
`IniHandler` (*class in kaptan.handlers.ini_handler*), 14

J

`JsonHandler` (*class in kaptan.handlers.json_handler*), 14

K

`Kaptan` (*class in kaptan*), 13, 14
`kaptan` (*module*), 13, 15, 17

L

`load()` (*kaptan.handlers.BaseHandler method*), 14
`load()` (*kaptan.handlers.dict_handler.DictHandler method*), 14
`load()` (*kaptan.handlers.ini_handler.IniHandler method*), 14
`load()` (*kaptan.handlers.json_handler.JsonHandler method*), 14
`load()` (*kaptan.handlers.yaml_handler.YamlHandler method*), 14

M

`main()` (*in module kaptan*), 14

U

`upsert()` (*kaptan.Kaptan method*), 13

Y

`YamlHandler` (*class in kaptan.handlers.yaml_handler*), 14