Hierarchical Conf Documentation

Release 1.0.0

QuintoAndar

Jun 14, 2023

CONTENTS

Made with by the **Data Engineering** team from QuintoAndar.

A library for loading configurations (or other metadata) hierarchically based on the current environment.

CHAPTER

ONE

HOW TO USE

Short

An example of how to use the library getting configurations:

```
from hierarchical_conf.hierarchical_conf import HierarchicalConf

hierarchical_conf = HierarchicalConf(searched_paths=[PROJECT_ROOT])
my_config = hierarchical_conf.get_config("my_config_key")
```

Long

This tool retrieve the configurations from (YAML) files according to the current environment and files precedence.

It receives a list of paths and searches each one for environment configuration files in an **orderly fashion**, loading them when found and **overwriting duplicated** configuration keys by the value of the key available in the file loaded at last. The YAML configuration files are expected to be named with prefixes based on the working environment, retrieved by the value of a pre-existent operational system environment's variable named `ENVIRONMENT`.

E.g.: Given the respective environments `dev` and `production` configuration files below:

dev_conf.yml:

```
foo: bar_dev
foo2: bar_dev2
```

production_conf.yml:

```
foo: bar_prod
foo2: bar_prod2
```

and given we are at development environment (`ENVIRONMENT=dev`), the following code will load the configuration file from the development environment file (`/my_path/dev_conf.yml`).

```
hconf = HierarchicalConf(conf_files_paths=['/my_path/'])
foo_conf = hconf.get_config("foo")
print(foo_conf)
# prints: bar_dev
```

Given `ENVIRONMENT=production`, the code above will load the configuration file from the production environment file (`/my_path/production_conf.yml`) and print: `bar_prod`.

To learn more use cases in practice (and about the keys overwriting), see Hierarchical Conf examples.

1.1 Navigation

1.1.1 Getting Started

Hierarchical Conf depends on **Python 3.7+**.

Python Package Index hosts reference to a pip-installable module of this library, using it is as straightforward as including it on your project's requirements.

```
pip install hierarchical-conf
```

Or after listing `hierarchical-conf` in your `requirements.txt` file:

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

Discovering Hierarchical Conf

Click on the following links to open the examples:

#1 Get configurations using one file

#2 Get configurations using two (or more) files

#3 Get specific configurations (runnable)

#4 Get specific configurations with overload (runnable)

Available methods

The features of this package can be accessed through:

- `configs`
- `get_config`

1.1.2 API Specification

hierarchical_conf package

Submodules

Module contents

Top-level package for hierarchical_conf API Client Python.

PYTHON MODULE INDEX

h

INDEX