
opsas-pylib

Release 0.2.2

calmzhu

May 24, 2020

CONTENTS

1 说明	1
2 安装	3
3 模块	5
3.1 config client	5
3.2 SlackLogHandler	6
Python Module Index	9
Index	11

说明

运维脚本写的多。工程方面有些不足。因此用这个项目，了解一下一个好的工程应该有代码之外的东西

比如：

- 使用文档
- 测试用例
- 自动化部署
- 不同环境的配置的管理
- 敏感信息的安全存储

另外，写完上面这些，其实可以帮助理清思路，代码的目的是什么，不能只限于无头绪的自娱自乐。

安装

```
pip install opsas-pylib
```


3.1 config client

3.1.1 应用场景

在接手运维一些遗留项目时，发现一个配置地狱。项目时间长或者经手人多了之后。

由于：

- 可能会同时从环境变量，配置文件，配置中心客户端等同时获取配置
- 然后配置之间有优先级覆盖关系

导致在多个环境或者线上 trouble shooting 时，实际生效的配置项及配置项来源不够可视化。因此借照 springboot application.yaml 的形式。将一个项目使用到的所有配置放在 jinja2 渲染的 yaml 文件中。这样部署维护就非常清晰了

3.1.2 使用说明

class opsas.utils.ConfigClient (*config_path*, *logger=None*)

render a key-value pair from config file. config file was yaml format with jinja2 support.

Parameters

- **config_path** (*str*) –Template config file path
- **logger** (*logging.logger*, *optional*) –Python logger object

Note: os environment can be used in yaml via context os_env

Example

```
>>> import os
>>> os.environ.setdefault('env', 'test')
>>> configClient = ConfigClient("config.yaml")
## config.yaml
env: {{ os_env.get("env") | default("local", True) }}
>>> configClient.get('env')
'test'
```

get (*k*)

Get config value via key

3.2 SlackLogHandler

python log handler, 用于将日志打到 slack

3.2.1 应用场景

运维系统的一些消息时希望通知到开发的。一般是整个发布完成后再抓去处理日志再用邮件等通知开发。但这样不利于异常处理。希望把发布的一些信息实时的发布出来供开发订阅。

3.2.2 使用说明

class opsas.utils.SlackLogHandler (*token, channel, logger=None*)

SlackLogHandler for python logging.logger object

Parameters

- **token** (*str*) -slack app oauth token
- **channel** (*str*) -slack channel name
- **logger** (*logging.logger, optional*) -logger object for logging logs when slack-loghandler,

Note: This slacklog handler used slack app to sendmessage.

Create a slack app and follower guides to allow it to connect and chat in channel https://api.slack.com/apps?new_app=1.

Examples

```
>>> import logging
>>> logger = logging.getLogger('slack')
>>> logging.basicConfig()
>>> slackLogHandler = SlackLogHandler(channel='test',token='xxx')
>>> logger.addHandler(slackLogHandler)
>>> logger.info("info")
>>> logger.warning("warnning")
```

```
>>> import logging
>>> logging.basicConfig()
>>> logger = logging.getLogger("slack")
>>> slackLogHandler = SlackLogHandler(channel='test',token='xxx')
>>> slackLogHandler.create_session('testDialog')
>>> logger.addHandler(slackLogHandler)
>>> logger.warning('warining')
```

create_session (*name*)

By default,logs were send to slack channel directly. Can also start a dialog in slack channel and all logs send as reply to this message :param name: dialog name :type name: str

Returns

Return type None

- search

PYTHON MODULE INDEX

O

opsas, [5](#)

INDEX

C

`ConfigClient` (*class in opsas.utils*), [5](#)

`create_session()` (*opsas.utils.SlackLogHandler*
method), [7](#)

G

`get()` (*opsas.utils.ConfigClient method*), [6](#)

O

`opsas` (*module*), [5](#)

S

`SlackLogHandler` (*class in opsas.utils*), [6](#)