turf Documentation

Release

Hurricane Labs, LLC

July 13, 2016

Contents

1 turf package **1**
- 1.1 Submodules . . . 1
- 1.2 turf.config module . . . 1
- 1.3 Module contents . . . 1

2 turf **3**

3 Get Turf **5**

4 Run the tests **7**

5 Basic Configuration Manager **9**

6 S3 Config Example IAM Policy **11**

CHAPTER 1

turf package

1.1 Submodules

1.2 turf.config module

1.3 Module contents

CHAPTER 2

turf

Configuration and environment inspection library

Turf makes managing configuration for your Python application easy. It provides a standard file access and naming convention, as well as extensive configuration validation. Turf allows you to specify what your applications configuration should look like, and then worry about your application and not parsing config files.

CHAPTER 3

Get Turf

CHAPTER 4

Run the tests

```
git clone git@github.com:HurricaneLabs/turf.git
cd turf
tox
```

CHAPTER 5

Basic Configuration Manager

```
$ cat /tmp/turftest/foo.yml
---
blah: bar
```

```
from turf.config import BaseConfig

class MyConfig(BaseConfig):
    config_dir = "/tmp/turftest"
    schema = {"foo":{"blah":{"type":"string"}}}

config = MyConfig()
print(config["foo"]["blah"])
```

Will produce:

```
bar
```

CHAPTER 6

S3 Config Example IAM Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::my-app-config",
                "arn:aws:s3:::my-app-config/*"
            ],
            "Principal": {
                "AWS": [
                    "arn:aws:iam:::role/my-app-role"
                ]
            }
        }
    ]
}
```