KBAC - Key-based Access Control

Introduction

Provides simple access control based on key+principals (all strings).

Examples

By role:

```
import kbac

ac = kbac.KBAC()
ac.add("viewer", ["bob", "joe"])
ac.add("manager", ["lucy", "rick"])

print(ac.is_allowed(["viewer"], ["lucy"]))
print(ac.is_allowed(["manager"], ["lucy"]))
print(ac.is_allowed(["viewer", "manager"], ["rick"]))
```

Notes:

- add individual settings using `KBAC.add()`
- test if lucy is a viewer
- test if lucy is a manager
- test if rick is a viewer or manager

By service:

```
import kbac

acdict = {
    "list": "dudley, al, jeffrey",
    "create": "dudley, jill",
    "delete": "dudley",
}

ac = kbac.KBAC()
ac.load(acdict.items())

principals = ["dudley"]
print "principals (%s)" % (principals,)
for key in acdict:
    print("key (%s) allowed (%s)" % (key, ac.is_allowed([key],
principals)))
```

Notes:

- load KBAC object from dictionary
- test which keys/services dudley is allowed

Display settings:

About

Requirements:

Python

License:

BSD-3

Repository:

https://bitbucket.org/johnmdev/kbac

Email:

expldotinfo@gmail.com

Activity

conflcmd - Manage Confluence Content
Aug 02, 2019 • updated by John • view change

Navigation Link Macro for Confluence
May 11, 2019 • updated by John • view change

KBAC - Key-based Access Control
Mar 10, 2019 • updated by John • view change

ThreadPool
Mar 10, 2019 • updated by John • view change

Miscellany
Mar 10, 2019 • updated by John • view change

Go-Flavored Error Handling in Python
Mar 02, 2019 • updated by John • view change

_sidebar
Mar 02, 2019 • updated by John • view change

_shortcuts
Mar 02, 2019 • updated by John • view change

0001-Support-for-slot-multiplier.patch
Aug 31, 2018 • attached by John

Slot Multiplier for Calculating CPU Usage in Gridengine
Aug 31, 2018 • updated by John • view change

```
import kbac

acdict = {
    "list": "dudley, al",
    "create": "dudley, brittany",
    "delete": "dudley",
}

ac = kbac.KBAC()
ac.load(acdict.items())

for key in ac.keys():
    print("key (%s) principals (%s)\n" % (key, sorted(ac.principals(key))))
```

Notes:

- extract settings using `KBAC.keys()` and `KBAC.principals()`