ipython$_{s}ecrets$Documentation

Release 1.1.1

Oliver Steele

Jun 15, 2018

Contents:

IPython Secrets makes it easier to use secrets in a Jupyter notebook.

The first time *get_secret* is called, it prompts the user for the password or other secret. After this value is read, it is saved in the system keyring, and the Jupyter output cell is then cleared.

Subsequent calls to `get_secret` use the saved value. This is true even if they are called in a different process running in the same account – for example, if the machine has been rebooted, or the local Jupyter server has otherwise been shut down and restarted.

Note: Install the gsheet-keyring package to use ipython-secrets on Google Colaboratory: `pip3 install gsheet-keyring`. The gsheet-keyring package provides a Keyring backend that is backed by Google Sheets. This is necessary since the Colaboratory environment doesn't provide a persistent file system, or the OS services that the Keyring's standard and alternative backends rely on.

Warning: This package stores the secret on the Jupyter server. Don't trust it with secrets you don't trust the Jupyter server with. (This is true for all means of using a secret in a notebook.)

Warning: The package is intended to reduce the likelihood of accidental disclosure of secrets in notebook source. It won't secure a secret from code that is running *in* the notebook; and it won't keep you from writing code that displays the secret in a notebook output cell – in which case it has been disclosed to whoever can see the notebook.

CHAPTER 1

Install

```
pip3 install ipython-secrets
```

CHAPTER 2

Usage

```
from ipython_secrets import *

TWILIO_API_KEY = get_secret('TWILIO_API_KEY')
```

CHAPTER 3

Alternatives

Secrets can also be stored in an environment variable, and read from the notebook. This is a best practice for applications (and especially web and other server-side services), but I've found it inconvenient for notebooks – the notebook server must be re-started to pick up a new environment variable; and, it complicates the setup instructions for notebook users.

CHAPTER 4

Development

Install Pipenv, and required packages:

```
$ pip3 install pipenv
$ pipenv install
$ pipenv shell
$ pip install flit
```

Install locally:

```
flit install --symlink
```

CHAPTER 5

Acknowledgements

This package is a thin wrapper around Keyring.

CHAPTER 6

License

MIT

CHAPTER 7

API

This package provides functions for using secrets in a Jupyter notebook.

These functions are for use in a notebook that needs to make use of secrets, such as passwords and API keys, to avoid storing the secret in the notebook source.

Note: This package uses Keyring. See the Keyring API documentation for additional information about where secrets are stored, and how to change the default location.

ipython_secrets.**get_secret**(*servicename*, *, *username=None*, *default=<object object>*, *force_prompt=False*, *prompt=None*)

Read a stored secret, or prompt the user for its value.

Look for a secret in the keyring. If it's not present, prompt the user, clear the cell, and save the secret.

Parameters

- **servicename** (*str*) – A keyring service name.
- **username** (*str, optional*) – A keyring username. This defaults to the value of the USER environment variable. (Note that this can programmatically altered.)
- **default** (*str, optional*) – The default value, if the secret is not present in the keyring. If this is supplied, the user is never prompted.
- **force_prompt** (*str, optional*) – If true, the user is always prompted for a secret.
- **prompt** (*str, optional*) – The text displayed to the user as part of the prompt.

Examples

```
from ipython_secrets import *

TWILIO_API_KEY = get_secret('TWILIO_API_KEY')
TWILIO_API_KEY = get_secret('TWILIO_API_KEY', 'my-account')
```

(continues on next page)

(continued from previous page)

```
TWILIO_API_KEY = get_secret('TWILIO_API_KEY', 'my-account',
                            prompt="Enter the API key")
```

ipython_secrets.**set_secret**(*servicename*, *password*, *, *username=None*)

Set a secret value.

Parameters

- **servicename** (*str*) – A keyring service name.
- **password** (*str*) – A keyring service name.
- **username** (*str, optional*) – A keyring username. This defaults to the value of the USER environment variable.

Notes

The argument order to *set_secret* is different from `keyring.set_password()`, and *username* can only be used as keyword parameter. This is in order that *username* can be optional, for compatibility with the more-frequently-used functions in this package.

ipython_secrets.**delete_secret**(*servicename*, *username=None*)

Delete a secret from the keyring.

Parameters

- **servicename** (*str*) – A keyring service name.
- **username** (*str, optional*) – A keyring username. This defaults to the value of the USER environment variable.

Python Module Index

i

Index