

# Managing Configuration

---



**Paul O'Fallon**

DISTINGUISHED ENTERPRISE ARCHITECT, COX AUTOMOTIVE

@paulofallon



# Overview



Discuss the “what” and “where”

How to do this in Node.js

Security implications of sensitive data

Add some tests



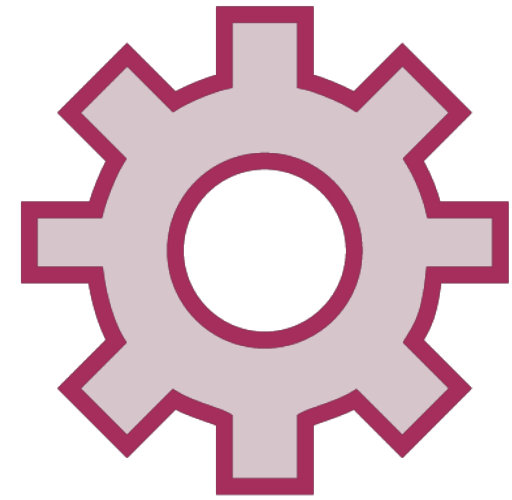
# A Variety of Configuration Information



Credentials



User preferences



Application settings

# Popular Configuration File Formats



Git config, Pypi config, AWS credentials



Visual Studio Code settings, Yeoman config



Maven settings, NuGet config





Typically scoped to the user

Stored in the user's home directory

In a file prefixed with a “dot”

Or in a directory prefixed with a “dot”



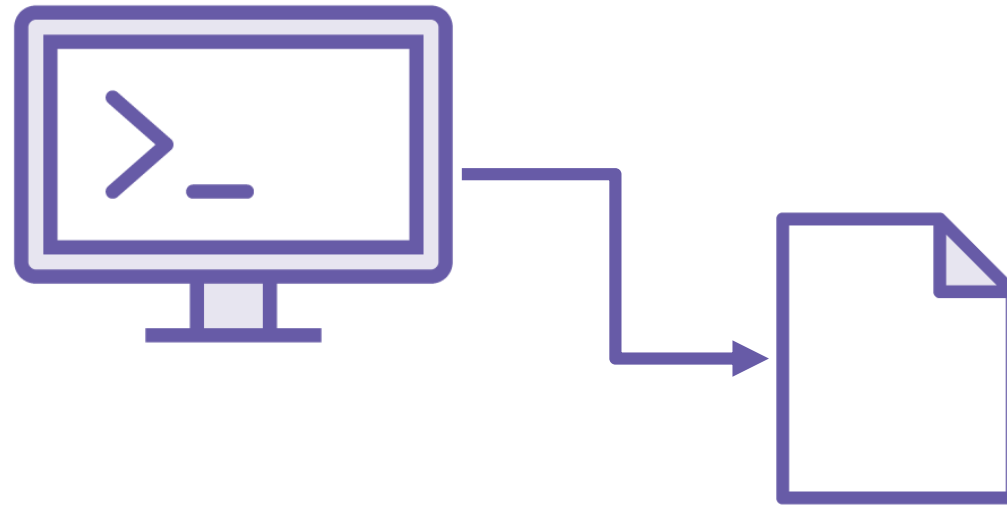
```
Command Prompt
Volume in drive C is Windows
Volume Serial Number is FE04-CA85

Directory of C:\Users\paul

12/06/2017  09:47 AM    <DIR>          .
12/06/2017  09:47 AM    <DIR>          ..
08/25/2017  01:52 PM    <DIR>          .android
08/28/2017  04:29 PM    <DIR>          .atom
07/23/2017  07:47 PM    <DIR>          .aws
06/03/2017  09:04 PM    0  .bashrc
06/03/2017  04:16 PM    <DIR>          .conda
09/06/2017  08:02 AM    <DIR>          .config
06/27/2017  07:41 PM    <DIR>          .dnx
10/03/2017  09:35 AM    <DIR>          .docker
10/09/2017  01:16 PM    <DIR>          .dotnet
09/03/2017  03:17 PM    <DIR>          .electron
12/06/2017  09:37 AM    187 .gitconfig
10/28/2017  01:11 PM    3,751 .hyper.js
09/06/2017  08:02 AM    <DIR>          .hyper_plugins
08/25/2017  01:51 PM    <DIR>          .IdeaIC2017.2
12/06/2017  09:47 AM    <DIR>          .IdeaIC2017.3
06/13/2017  02:47 PM    <DIR>          .keras
12/06/2017  04:12 PM    <DIR>          .m2
-- More --
```



# Managing Configuration in Node.js



# ★ configstore public

Easily load and persist config without having to think about where and how

Config is stored in a JSON file located in `$XDG_CONFIG_HOME` or `~/.config`.

Example: `~/.config/configstore/some-id.json`

If you need this for Electron, check out [electron-store](#) instead.

## Usage


```
const Configstore = require('configstore');
const pkg = require('./package.json');

// create a Configstore instance with a unique ID e.g.
// Package name and optionally some default values
const conf = new Configstore(pkg.name, {foo: 'bar'});


console.log(conf.get('foo'));
//=> 'bar'

conf.set('awesome', true);
console.log(conf.get('awesome'));
//=> true

// Use dot-notation to access nested properties
conf.set('bar.baz', true);
```

 `npm install configstore`

[how?](#) [learn more](#)

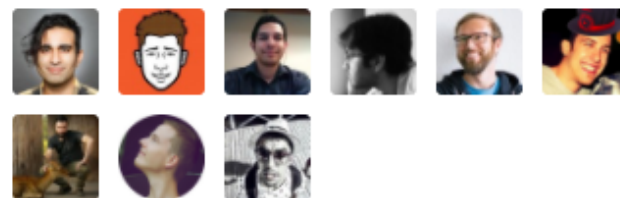
 [sindresorhus](#) published 5 months ago

**3.1.1** is the latest of 26 releases

[github.com/yeoman/configstore](https://github.com/yeoman/configstore)

BSD-2-Clause

Collaborators [list](#)



## Stats

**328,290** downloads in the last day

**1,897,938** downloads in the last week

**7,747,189** downloads in the last month

**2 open issues** on GitHub

**No open pull requests** on GitHub





## Usage

```
const Configstore = require('configstore');
const pkg = require('./package.json');

// create a Configstore instance with an unique ID e.g.
// Package name and optionally some default values
const conf = new Configstore(pkg.name, {foo: 'bar'});

console.log(conf.get('foo'));
//=> 'bar'

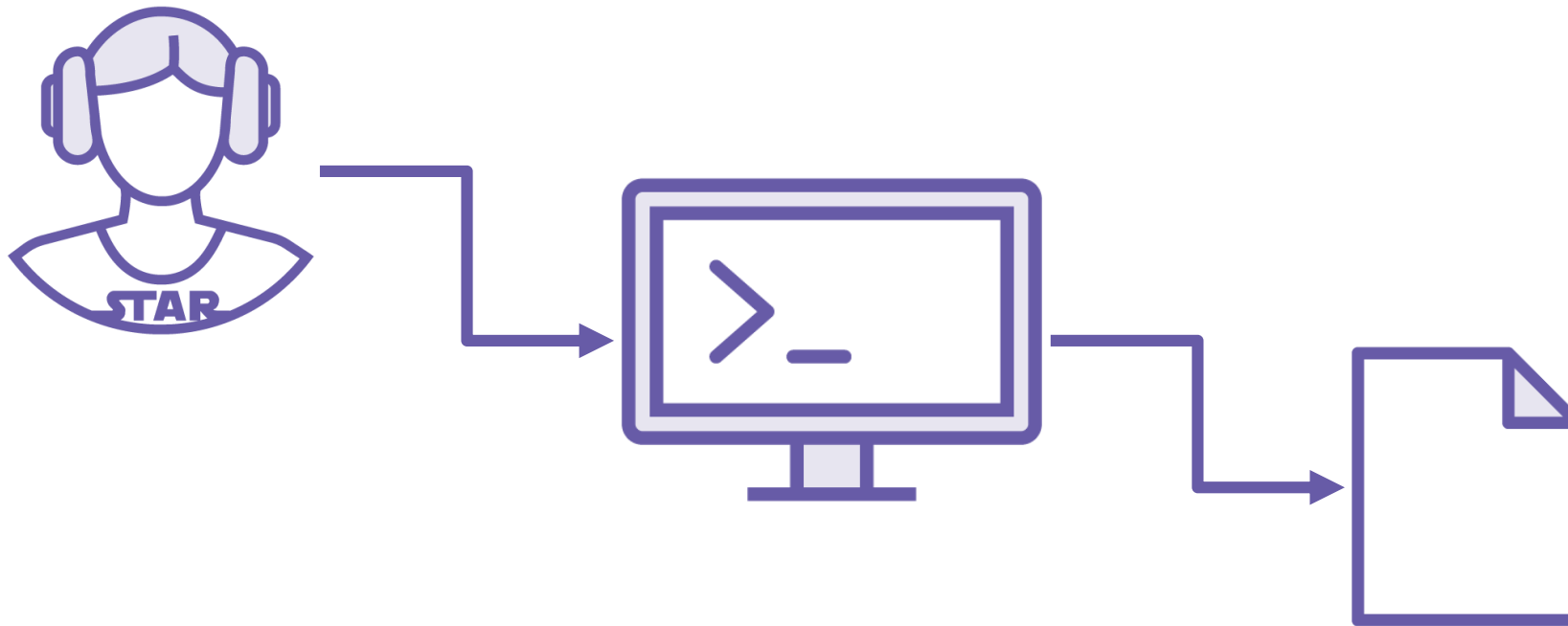
conf.set('awesome', true);
console.log(conf.get('awesome'));
//=> true

// Use dot-notation to access nested properties
conf.set('bar.baz', true);
console.log(conf.get('bar'));
//=> {baz: true}

conf.delete('awesome');
console.log(conf.get('awesome'));
//=> undefined
```



# Soliciting the User for Configuration Input



★ **inquirer** public

npm package 4.0.1 build passing coverage 96% dependencies up to date

A collection of common interactive command line user interfaces.

**Version 4.x** only supports Node 6 and over. For Node 4 support please use **version 3.x**.

## Table of Contents

1. **Documentation**
2. **Installation**
3. **Examples**
4. **Methods**
5. **Objects**
6. **Questions**
7. **Answers**
8. **Separator**
9. **Prompt Types**
10. **User Interfaces and Layouts**
11. **Reactive Interface**
12. **Support**
13. **News**
14. **Contributing**

📦 npm install inquirer

[how? learn more](#)

 **sboudrias** published a week ago

**4.0.1** is the latest of 74 releases

[github.com/SBoudrias/Inquirer.js](https://github.com/SBoudrias/Inquirer.js)

MIT

Collaborators [list](#)



## Stats

**636,505** downloads in the last day

**3,460,805** downloads in the last week

**14,617,143** downloads in the last month

**82 open issues** on GitHub

**4 open pull requests** on GitHub



## Documentation

---

### Installation

```
npm install inquirer
```

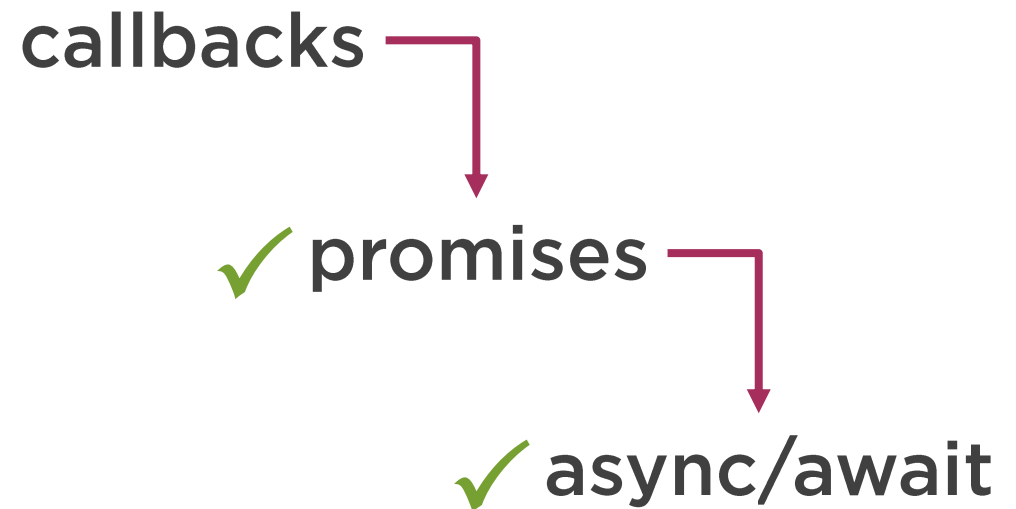
```
var inquirer = require('inquirer');  
inquirer.prompt([/* Pass your questions in here */]).then(answers => {  
  // Use user feedback for... whatever!!  
});
```



# Node.js Features We'll Be Using

## ES6 Classes

```
class MyClass {  
  constructor() {  
    // ...  
  }  
  
  doIt() {  
    // ...  
  }  
}  
  
var s = new MyClass()  
s.doIt()
```



# Demo



Introduce a Credential Manager

Retrieve our API key and secret

Prompt the user if these don't exist



# Testing Our Command Line Application



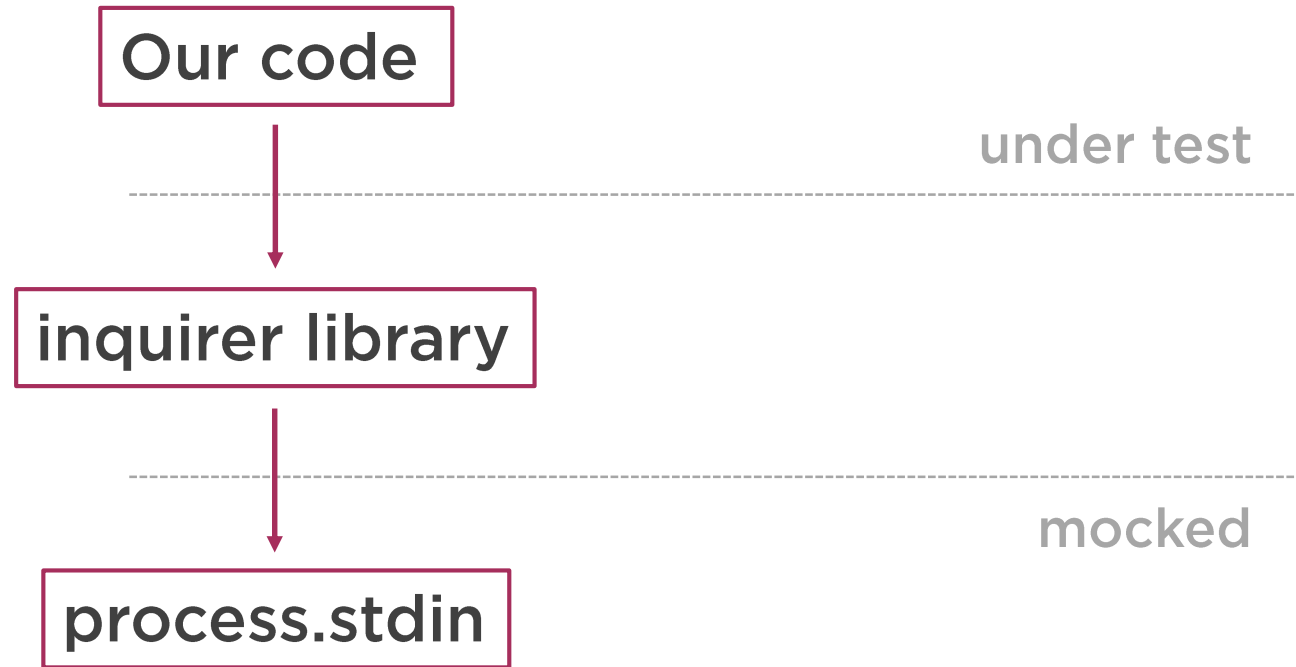
**Test Framework: Mocha** (mochajs.org)

**Assertion Library: Chai** (chaijs.com)

**Mocks: Sinon** (sinonjs.org)

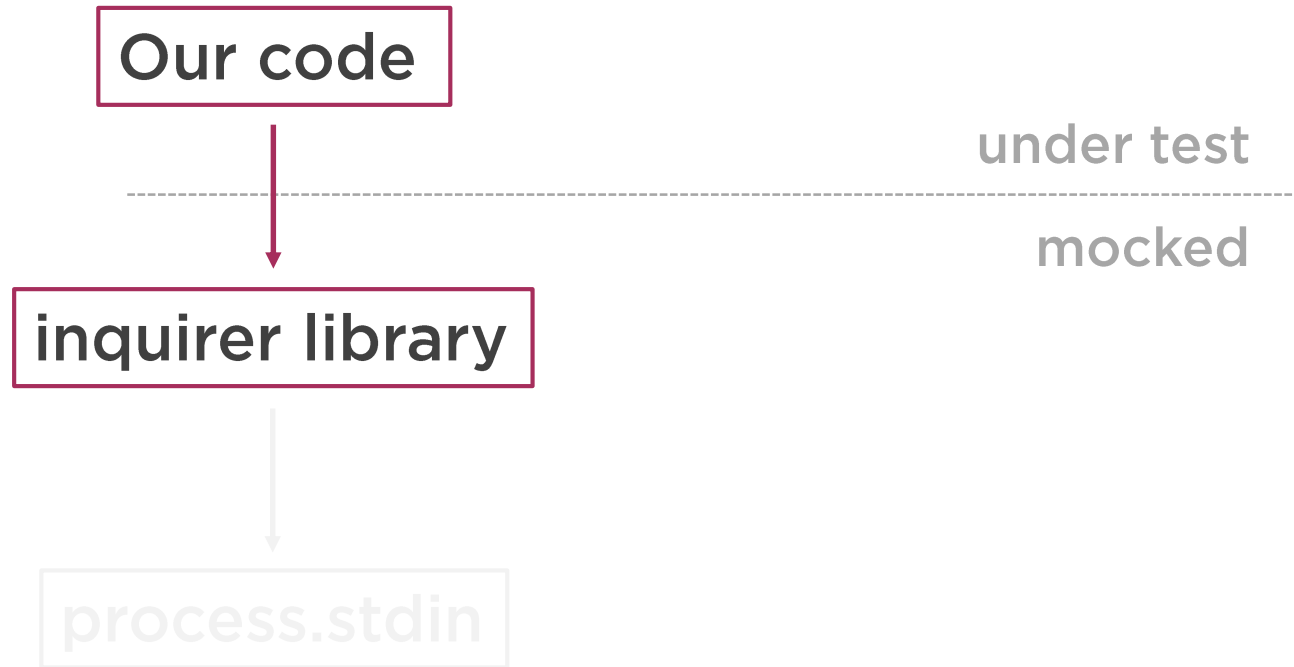


# Mocking User Input for a CLI Test





# Mocking User Input for a CLI Test



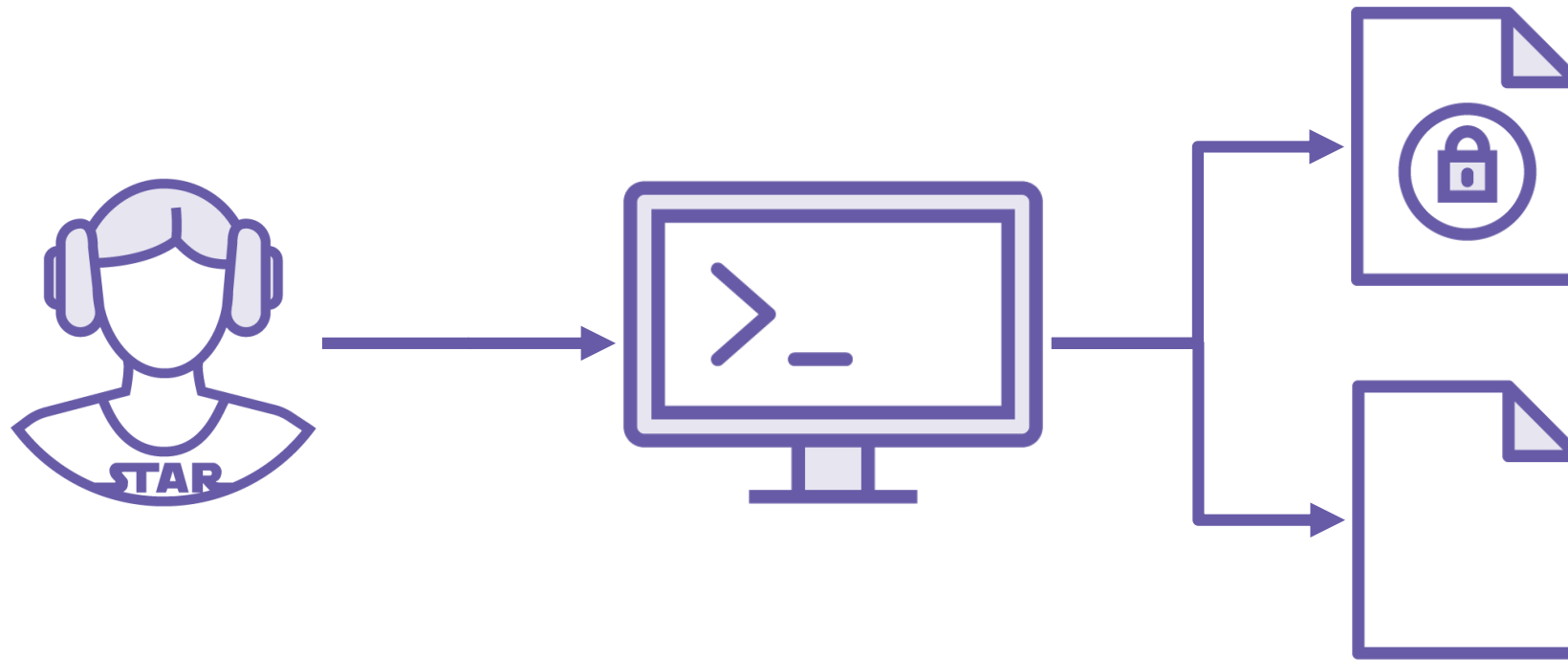
# Demo



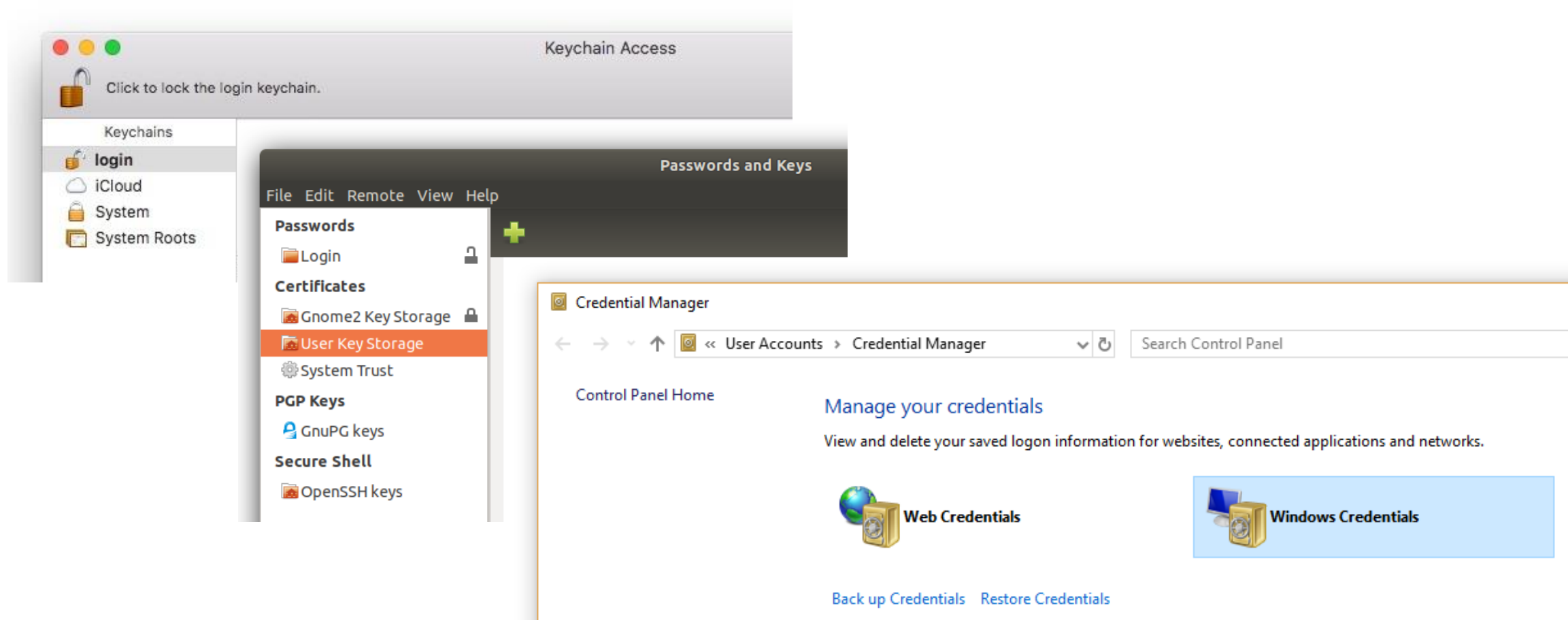
Add a test for our Credential Manager



# Securing Our API Secret



# Securely Storing Credentials





# keytar

publicbuild passingbuild passingdependenciesup to date

A native Node module to get, add, replace, and delete passwords in system's keychain. On macOS the passwords are managed by the Keychain, on Linux they are managed by the Secret Service API/libsecret, and on Windows they are managed by Credential Vault.

## Installing

```
npm install keytar
```

### On Linux

Currently this library uses `libsecret` so you may need to install it before running `npm install`.

Depending on your distribution, you will need to run the following command:

- Debian/Ubuntu: `sudo apt-get install libsecret-1-dev`
- Red Hat-based: `sudo yum install libsecret-devel`
- Arch Linux: `sudo pacman -S libsecret`

## Building

- Clone the repository
- Run `npm install`
- Run `npm test` to run the tests



`npm install keytar`

[how?](#) [learn more](#)



[damieng](#) published a month ago

**4.0.5** is the latest of 30 releases

[github.com/atom/node-keytar](https://github.com/atom/node-keytar)

[atom.github.io/node-keytar](https://atom.github.io/node-keytar)

MIT

Collaborators [list](#)



### Stats

**1,347** downloads in the last day

**8,189** downloads in the last week

**38,747** downloads in the last month

**24 open issues** on GitHub

**5 open pull requests** on GitHub



# Demo



Securely store the API secret



# Summary



Types of configuration information

User-centric storage of config data

Added configstore to our application

Securely stored our secret with keytar

Tests!

