

Using a Package



Mosh Hamedani
programmingwithmosh.com

ActivitiesVisual Studio Code

Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

> 01. Introduction

> 02. Package.json

> 03. Installing a Node Package / ...

> node_modules / underscore

> LICENSE

> package.json

> README.md

> underscore-min.js

> underscore-min.js.map

> underscore.js

> package-lock.json

> package.json

> 04. Using a Package / npm-demo

> node_modules

> package-lock.json

> package.json

> 05. Package Dependencies

> 06. NPM Packages and Source Control

> 07. Semantic Versioning

> 08. Listing the Installed Packages

> 09. Viewing Registry Info for a Pack...

> 10. Installing a Specific Version of a ...

> 11. Updating Local Packages

> 12. DevDependencies

> 13. Uninstalling a Package

> 14. Working with Global Packages

> 15. Publishing a Package

> 16. Updating a Published Package

> 17. NPM Recap

OUTLINE

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

3: bash

dandy@My-Ubuntu:~/Keep/Node.js/03. Node Package Manager/Applications/04. Using a Package/npm-demo\$ sudo npm init --yes

[sudo] password for dandy:

Wrote to /home/dandy/Keep/Node.js/03. Node Package Manager/Applications/04. Using a Package/npm-demo/package.json:

{

"name": "npm-demo",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC"

}

dandy@My-Ubuntu:~/Keep/Node.js/03. Node Package Manager/Applications/04. Using a Package/npm-demo\$ sudo npm i underscore

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN npm-demo@1.0.0 No description

npm WARN npm-demo@1.0.0 No repository field.

+ underscore@1.9.2

added 1 package from 1 contributor and audited 1 package in 1.729s

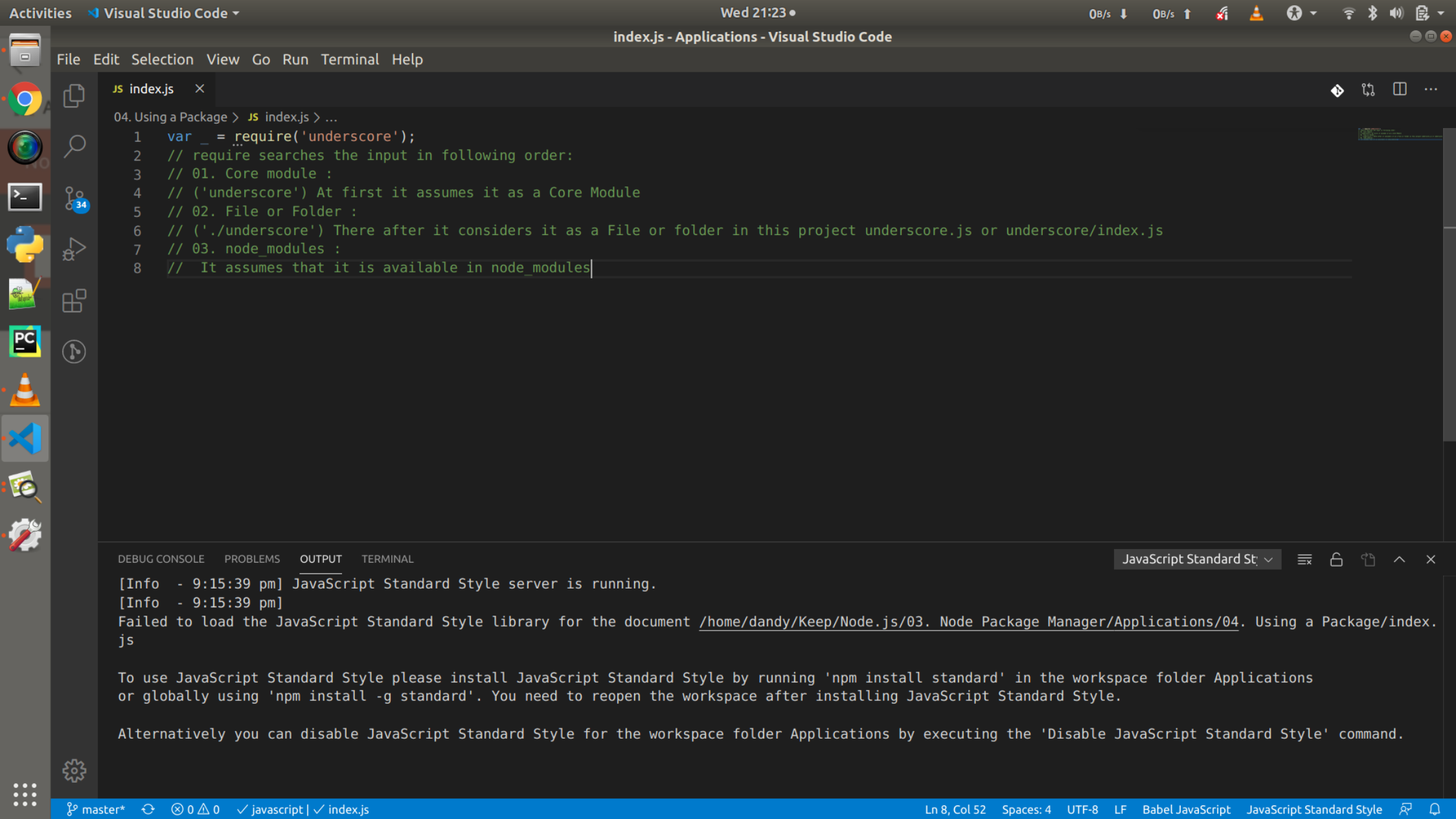
found 0 vulnerabilities

dandy@My-Ubuntu:~/Keep/Node.js/03. Node Package Manager/Applications/04. Using a Package/npm-demo\$

master*

0 0 0

✓ json | package.json



Underscore.js

underscorejs.org

Apps

For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Underscore.js (1.8.3)

» GitHub Repository

» Annotated Source

» Underscore-contrib

Filter

Introduction

Collections

- each
- map
- reduce
- reduceRight
- find
- filter
- where
- findWhere
- reject
- every
- some
- contains
- invoke
- pluck
- max

UNDERSCORE.JS

Underscore is a JavaScript library that provides a whole mess of useful functional programming helpers without extending any built-in objects. It's the answer to the question: "If I sit down in front of a blank HTML page, and want to start being productive immediately, what do I need?" ... and the tie to go along with [jQuery](#)'s tux and [Backbone](#)'s suspenders.

Underscore provides over 100 functions that support both your favorite workaday functional helpers: **map**, **filter**, **invoke** — as well as more specialized goodies: function binding, javascript templating, creating quick indexes, deep equality testing, and so on.

A complete [Test Suite](#) is included for your perusal.

You may also read through the [annotated source code](#).

Enjoying Underscore, and want to *turn it up to 11*? Try [Underscore-contrib](#).

Udacity

Underscore.js

underscorejs.org/#contains

☆

Apps For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Underscore.js (1.8.3)

» GitHub Repository

» Annotated Source

» Underscore-contrib

Filter

Introduction

Collections

- each
- map
- reduce
- reduceRight
- find
- filter
- where
- findWhere
- reject
- every
- some
- contains
- invoke
- pluck
- max

contains `_.contains(list, value, [fromIndex])` *Alias: **includes***

Returns *true* if the **value** is present in the **list**. Uses **indexOf** internally, if **list** is an Array. Use **fromIndex** to start your search at a given index.

```
_.contains([1, 2, 3], 3);  
=> true
```

invoke `_.invoke(list, methodName, *arguments)`

Calls the method named by **methodName** on each value in the **list**. Any extra arguments passed to **invoke** will be forwarded on to the method invocation.

```
_.invoke([[5, 1, 7], [3, 2, 1]], 'sort');  
=> [[1, 5, 7], [1, 2, 3]]
```

pluck `_.pluck(list, propertyName)`

A convenient version of what is perhaps the most common use-case for **map**: extracting a list of property values.

```
var stooges = [{name: 'moe', age: 40}, {name: 'larry', age: 50}, {name: 'curly', age: 60}];  
_.pluck(stooges, 'name');
```

Stoogy

