
poi Documentation

Release 1...1

Kalle Saari

October 02, 2016

1	Introduction	3
1.1	Design principles	4
2	Tutorial	5
3	Manual	9
3.1	Installation and setup	9
3.2	Configuration file	9
3.3	poi commands	10
4	API	11
4.1	NoteManager	11
4.2	UI	11
5	Indices and tables	13

Contents:

Introduction

`poi` is a command line tool for creating and managing notes written in plain text. It minimizes the hassle of naming, saving, and searching for notes. `poi` also provides a minimal markup language and syntax to display notes in a terminal screen with highlighting and active links. It is designed for notes of short to medium length.

Here is an example of a `poi` note in an editor. It uses highlighting, tags, and links:

```
Correlation versus independence
```

```
__Correlation measures linear relationship__ between random variables, while
__dependency measures any kind of relation__.
```

```
Thus independence implies zero correlation and correlation implies dependency,
but not vice versa. For example, let A and B be random variables such that
```

```
* P(A = 1) = P(A = -1) = 0.5
* P(B = 1000 | A = 1) = P(B = -1000 | A = 1) = 0.5
* P(B = 0 | A = -1) = 1
```

```
Then, intuitively, there is a strong relation between A and B, so they are not
independent. However, the relationship is not linear, so their correlation is 0
The above statements can also be derived algebraically.
```

```
[1] http://stats.stackexchange.com/questions/113417/does-non-zero-correlation-imply-dependence
[2] http://math.stackexchange.com/questions/444408/zero-correlation-does-not-imply-independence
```

```
#: statistics, example
```

This is what the note looks like when it is displayed by `poi` in the iTerm terminal:

```
2016-02-29-216
Correlation versus independence

Correlation measures linear relationship between random variables, while
dependency measures any kind of relation.

Thus independence implies zero correlation and correlation implies dependency,
but not vice versa. For example, let A and B be random variables such that

* P(A = 1) = P(A = -1) = 0.5
* P(B = 1000 | A = 1) = P(B = -1000 | A = 1) = 0.5
* P(B = 0 | A = -1) = 1

Then, intuitively, there is a strong relation between A and B, so they are not
independent. However, the relationship is not linear, so their correlation is 0
The above statements can also be derived algebraically.

[1] http://stats.stackexchange.com/questions/113417/does-non-zero-correlation-imply-dependence
[2] http://math.stackexchange.com/questions/444408/zero-correlation-does-not-imply-independence
statistics, example
> |
```

The green URLs in above image are active: you can open the second URL, for example, in your default browser by typing `o 2`. The top-most line is the automatically generated id for the note; it can be used to link to this note from another note. The last line lists the note's tags.

`poi` also supports storing and linking to arbitrary files from notes. As long as these files are stored in a designated directory, they can be linked to from a `poi` note and quickly opened with single key stroke.

1.1 Design principles

A guiding principle of `poi` is **simplicity**; the user interface of `poi` is very simple and quickly learned. The markup syntax and the structure of a `poi` note is intuitive, so there is hardly any need from memorization. However, `poi` also allows a small number of **configuration** options so you can use your favorite text editor and colors for highlighting.

Another guiding principle is that `poi` only helps the user to take better, faster, and prettier notes, not to lock the user in to using the program, which is usually the case in proprietary note taking software. Thus, for example, no database software or proprietary file formats are used. All information added by `poi` is stored in the filename in a human readable form such as:

```
2016-09-14-3-2016-10-03.txt
```

This means that the file was created on 2016-09-14, it was the third note created on that date, and it was last edited on 2016-10-03. Coincidentally, the prefix 2016-09-14-3 of the filename is the identifier of the note.

The final, and perhaps most important design principle, is the tried-and-tested Unix principle: do one thing and do it well. This application will not send tweets or train neural networks, it manages text notes, and it does it well.

The code is open source, MIT licenced.

Note: As of now, `poi` only works in Mac OS X, but the limitation is not serious. With someone's help, `poi` should be easily extendable to work in Windows and Linux.

Tutorial

poi is easy to learn, and the fastest way to do it is to follow an example session, so let's do that.

Let us first download and setup poi:

```
$ pip install -U poi
$ poi-config
poi installation directory [/Users/<user>/poi]:
<enter>
poi: created directory: /Users/<user>/poi
poi: created directory: /Users/<user>/poi/notes
poi: created directory: /Users/<user>/poi/refs
poi: created directory: /Users/<user>/poi/.backups
poi: a new configuration has been created at /Users/<user>/poiconfig.json
```

Now we are ready to use poi:

```
$ poi
> <press tab>
```

poi launches and greets you with >. If you press the tab key, you'll get a listing of all possible commands. Since we do not have any notes yet, we will add one pressing a (as in "add note"):

```
> a
```

Now an external text editor opens up. By default, this is TextEdit. Add the following contents to the file:

```
A reasonably meaningless note.

#: misc, philosophy
```

The first line will serve as a title of your note. The line starting with #: is the tag line. There every token separated by commas are regarded as tags by poi and treated separately. Save this with your text editor and go back to poi. There type:

```
> s

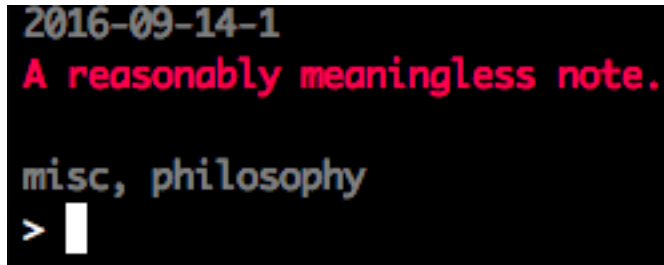
  0      2016-09-14 Wed      A reasonably meaningless note.

listing: 1              total: 1
```

In the listing above, the 0 is a dummy identifier that you can use to refer this particular note; it changes with every listing, so do get too attached to it. Next is the date, when the note was created (yours will be different from mine). Now press (d as in display):

```
> d 0
```

You should see something like this, except that the first line will show your date, not mine:



```
2016-09-14-1
A reasonably meaningless note.
misc, philosophy
> █
```

That is all good and well, but let's move on. Take any PDF file you happen to have and copy it to `<poi-root>/refs` and give it a name `quick brown fox.pdf`.

Next we will add another poi note. Again type:

```
> a
```

and create a note with the following content:

```
This is the title of my first poi note

Ths is te body of the text. I can highlight text in three different ways by

- using `backticks`,
- using :::triple colons:::, and
- using ___triple underscores___.

I can three different kinds of links:

1. URL [1],
2. a file in my ``refs`` directory [2], and
3. an identitier to another poi note [3].

In the very end of my note, I will first list my links and the list my tags
as follows:

[1] http://github.com
[2] quick brown fox.pdf
[3] 2016-09-14-1

#: test, poi, misc
```

Save this note, and return to poi. Press the following command after the prompt `>`. You should see an output identical to mine except for the dates:

```
> s

1    2016-09-14 Wed    A reasonably meaningless note.
0    2016-09-14 Wed    This is the title of my first poi note

listing: 2            total: 2

> s first

0    2016-09-14 Wed    This is the title of my first poi note
```

```

listing: 3          total: 1

> ss t
tag #1: philosophy
tag #2: <enter>

    0    2016-09-14 Wed    A reasonably meaningless note.

listing: 4          total: 1

> d 3:0

```

The last command should show you something like this:

```

2016-09-14-2
This is the title of my first poi note

Ths is te body of the text. I can highlight text in three different ways by

- using backticks,
- using triple colons, and
- using triple underscores.

I can three different kinds of links:

1. URL [1],
2. a file in my refs directory [2], and
3. an identitier to another poi note [3].

In the very end of my note, I will first list my links and the list my tags
as follows:

[1] http://github.com
[2] quick brown fox.pdf
[3] 2016-09-14-1
test, poi, misc
> 

```

Now if you press (o as in “open link”):

```
> o 1
```

Your web browser will show the URL for you. If you press:

```
> o 2
```

Your pdf reader will open and show you the PDF file. Finally if you press:

```
> o 3
```

poi will display the contents of the note with identifier 2016-09-14-1 (your identifier will be different).

This is the basic usage and work flow of poi: You add new notes with `a`, you search notes with `s` followed by keywords,

and you display highlighted notes in the terminal window with `d` followed by a listing number or an identifier. There are some more details to the capabilities of poi, and you can read about them in the manual, but you will get pretty far by just doing:

```
> <tab>
```

which gives a listing of all available commands in poi.

3.1 Installation and setup

Install poi:

```
$ pip install poi
```

Before you can start using poi, you have to give it a path to which install poi directory. The directory will have the following structure:

```
poi/  
  notes/  
  refs/  
  .backups/
```

By default, the root directory is `~`, but you may change it. The `notes` and `.backups` directories are best to leave alone, but feel free to put anything you like to the `refs` directory; items in `refs` can be referenced from a poi note.

These directories are created for you when you run:

```
$ poi-config
```

and answer a question about the installation directory. Running this command will also produce a configuration file, `.poiconfig.json` stored in your home directory.

3.2 Configuration file

The path of poi's configuration file is `~/.poiconfig.json`. You can edit this file manually as you like. Here is what is in it:

root The root directory for poi's directories. Default is `~/poi`

editor_cdm A command you run in command line to open an editor of your choice. Default is `open -a /Applications/TextEdit.app`. It is up to you to make sure that this is a valid command for launching the editor. The editor command can also be simply, say, `vim`, in which case the editor is launched in the terminal. The drawback is just that then the editor blocks access to poi, making it hard to work on multiple notes simultaneously.

highlight_markers These are three different markers with which you can add highlighting to your notes. A piece of text that is surrounded by two matching markers is highlighted when it is displayed on the terminal screen. By default, these markers are

- “\ `” (backtick)
- “: : :” (three colons)
- “_ _ _” (three underscores)

file_ext The file extension of a poi note. By default, it is `.txt`, but it you may want to change it so that a text editor can recognize it as a poi note and maybe add highlighting during the typing of a note.

color This is a dictionary defining the colors used in the highlighting of a note. By default it looks like

- "link": "#00ff66 bold",
- "title": "#ff0066 bold",
- "tag": "#808080 bold",
- "highlight": ["#0000ff bold", "#ffff00 bold", "#00ffff bold"]

This is self-explanatory. The entries in `highlight` match with the entries in `highlight_markers` above.

3.3 poi commands

TODO

This page is auto-generated API documentation. It is quite rough from the edges still, improvements are on the way. The main takeaway is that the class `note_manager.NoteManager` can be used to create multiple notetaking setups, e.g., one for personal notes and another one for word-related notes.

4.1 NoteManager

4.2 UI

Indices and tables

- `genindex`
- `modindex`
- `search`