Kanban-View for Things 3 - Release V1.1

Release note:

V1.1: Many thanks to Alexander Willner from Berlin, Germany for making the script fully compatible with both the stock 2.7 (coming with any recent macOS version) and 3.6 version of python. Furthermore, he introduce dynamic home folder as well as default output in the same directory as the script path. This removes the need of previous version to edit the script to change the unix user name in the file path. He also added an extra check in the database query to exclude projects that might be sitting in the trash folder.

Synopsis:
The following will allow a visual project-level overview of what is on your plate. It is comprised of 3 files: a python script, a CSS file and a PDF README file (a copy of this post). It can be executed from the terminal or as a system service using the macOS Automator app (thus never needing to open a terminal and can be associated to a keyboard short cut).

*** read below for the download link ***

Disclaimer:
I am a physicist not professional programmer. This means that the code work but by no means that it is the most efficient or nicest piece of scripting you have ever seen. In fact, at this point in time, I have made no effort to clean, to subdivide or to make the code more compact.

This script is provided as is. You are free to modify any file as you see fit. If you do use it and like it, please do share this post. If you do modify script and make it better, I would like to hear about / share it in the comment section!

What does it do:
The python script will extract the information in Things 3 SQLite3 database (assumed to reside in its std place on your hard drive. As such, you only need to change the username from beaulieu to yours and it should work!) and separate it in upcoming projects i.e. active projects (type=1 and status =0) with due dates(dueDate!=0), active projects that have no active tasks and finally, all other active projects.

It will then build on the fly an html file call KanbanView.htm, using the 3col.css file containing all of the style information, and automatically start your browser view of the file once completed (webbrowser.open_new_tab(filename))

Note: The CSS file is easy to modify to add one or more columns. I myself use a 4-column version, where I separate the Mentoring area projects (between 15 and 20) from the other upcoming projects.

What it is meant for:
Provide a visual cue of each project that is on your plate at the moment, in particular those with due dates. This is akin of something in-between the 10 000 and 20 000 feet altitude review in David Allan GTD book.

This is helpful in two aspects:
1- After the weekly review to make sure you did not forget something obvious (e.g. the No Active Task column should be empty!)

2- Anytime you need to make a decision about committing to a new project, have an overview of what is actually ON your plate. This will help you saying yes or, more importantly, saying no!

What it is NOT meant for:
It does not replace your task manager in the day-to-day operation. Things should be the place to be for that. More importantly, it does not replace the crucial weekly review. As written previously it could supplement it and provide a more visual and quick access to your active projects, but that is it. Finally, if you do have extremely complex projects, it does not replace something like OmniPlan or Microsoft Project.

Files included and where they should go:

DOWNLOAD files here: https://github.com/lucbeaulieu/KanbanView

1. Unzipping will show a folder that contains:
   - getThings3.py: the main python script
   - 3col.css: ccs style for on the fly html visualization style (column definition, color, independent scrolling, …)
   - README.pdf: self-explanatory!

2. Place the folder KanbanView at the location of your choice (I suggest your Document folder)

What you need on your Mac:
- Latest version of Things 3

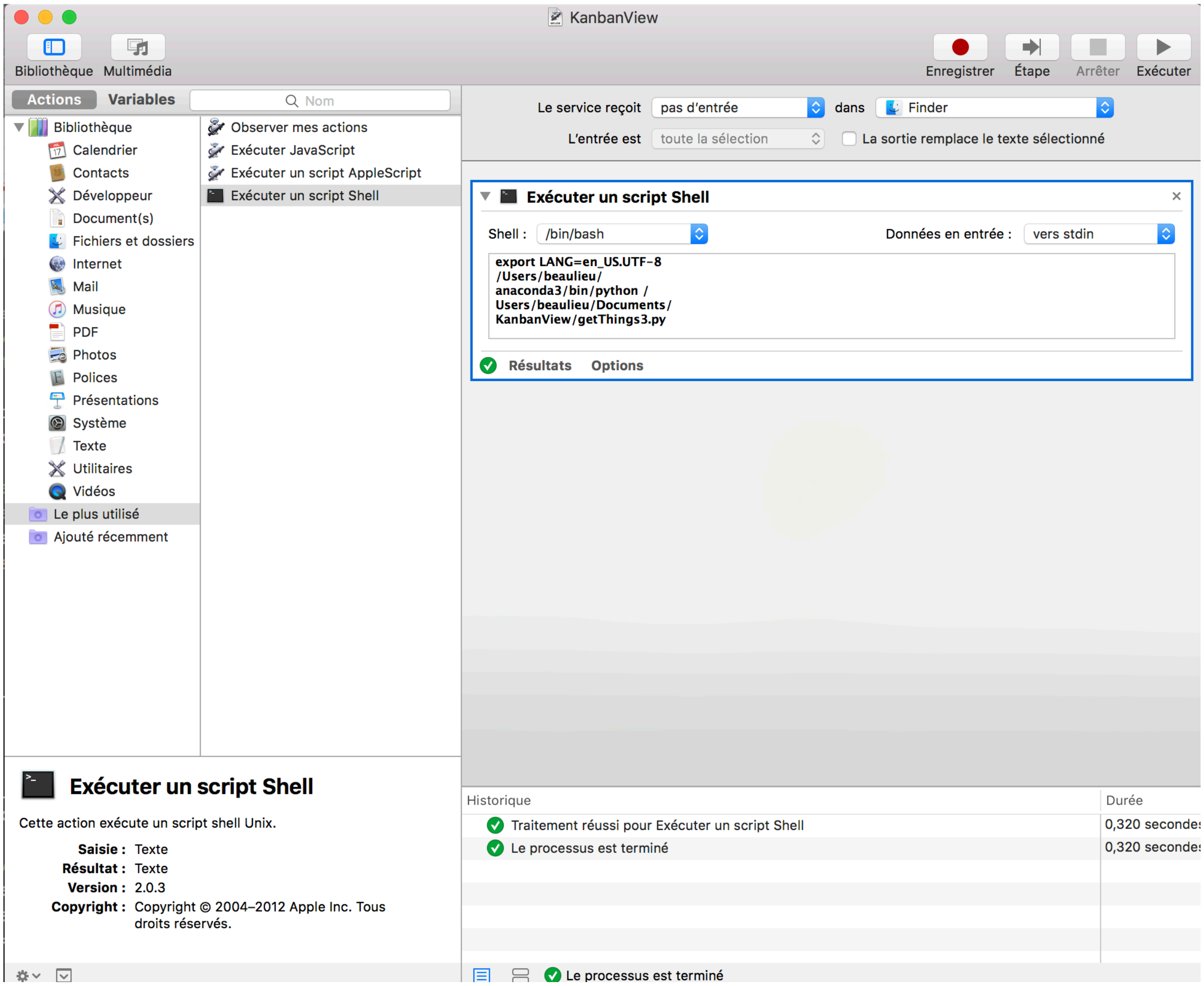
- Python 3.6 (it might run with python 2.7 that is installed by default on macOS (10.13 and earlier) but I have not tested it.
- Safari (I have not tested it with anything else but I am using only std html, nothing fancy)

To run - method 1 - the terminal:

- Open a terminal
- Move to your KanbanView folder: e.g. cd ./Documents/KanbanView assuming you save the folder where I told you earlier.
- Then simply do: python getThings3.py
- You will:
  - See your Kanban on your screen, in your favorite browser.
  - See also that each entry has a clickable link that open that given project back in Things. You will likely get a message box asking permission to Things to do it, just say yes and look at your Things 3 app!

To run - method 2 - set up as a System Service (never need to open a terminal or run python once it is installed!):

- Open the Automator app (in the Application folder)
- Chose « run shell script » as in the box below:



Le service reçoit: pas d'entrée dans Finder

L'entrée est: toute la sélection ☐ La sortie remplace le texte sélectionné

Exécuter un script Shell

Shell : /bin/bash Données en entrée : vers stdin

```
export LANG=en_US.UTF-8
/Users/beaulieu/
anaconda3/bin/python /
Users/beaulieu/Documents/
KanbanView/getThings3.py
```

Exécuter un script Shell

Cette action exécute un script shell Unix.

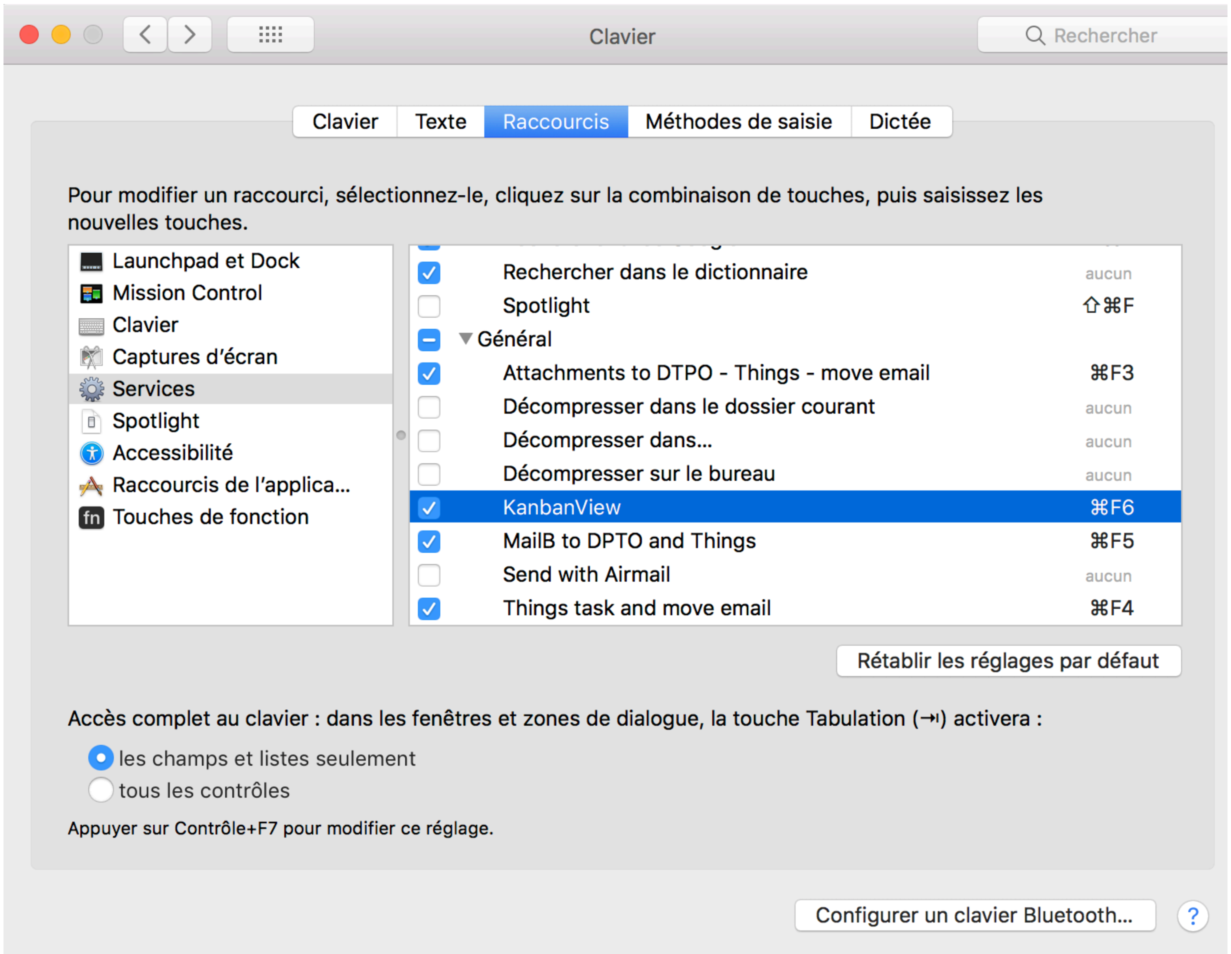
Saisie : Texte
Résultat : Texte
Version : 2.0.3
Copyright : Copyright © 2004–2012 Apple Inc. Tous droits réservés.

Historique	Durée
✓ Traitement réussi pour Exécuter un script Shell	0,320 seconde
✓ Le processus est terminé	0,320 seconde

✓ Le processus est terminé

In Automator, pick a new System service and then « run a shell script » with /bin/bash, stdin as input as shown. At the very top, the service received no input and select Finder. You see that the shell has only two lines. The first one set the text encoding. This is necessary because in Automator the default is different from that set in your system environment when running from the Terminal. The second line is simply running the script!

- Next, assign it to a keyboard combination so that you can call (or refresh) your Kanban view on the fly, as displayed below.



Clavier

Rechercher

Clavier | Texte | Raccourcis | Méthodes de saisie | Dictée

Pour modifier un raccourci, sélectionnez-le, cliquez sur la combinaison de touches, puis saisissez les nouvelles touches.

- Launchpad et Dock
- Mission Control
- Clavier
- Captures d'écran
- Services
- Spotlight
- Accessibilité
- Raccourcis de l'applica...
- Touches de fonction

☑	Rechercher dans le dictionnaire	aucun
☐	Spotlight	⇧⌘F
⊟	▼ Général	
☑	Attachments to DTPO - Things - move email	⌘F3
☐	Décompresser dans le dossier courant	aucun
☐	Décompresser dans...	aucun
☐	Décompresser sur le bureau	aucun
☑	KanbanView	⌘F6
☑	MailB to DPTO and Things	⌘F5
☐	Send with Airmail	aucun
☑	Things task and move email	⌘F4

Rétablir les réglages par défaut

Accès complet au clavier : dans les fenêtres et zones de dialogue, la touche Tabulation (⇥) activera :

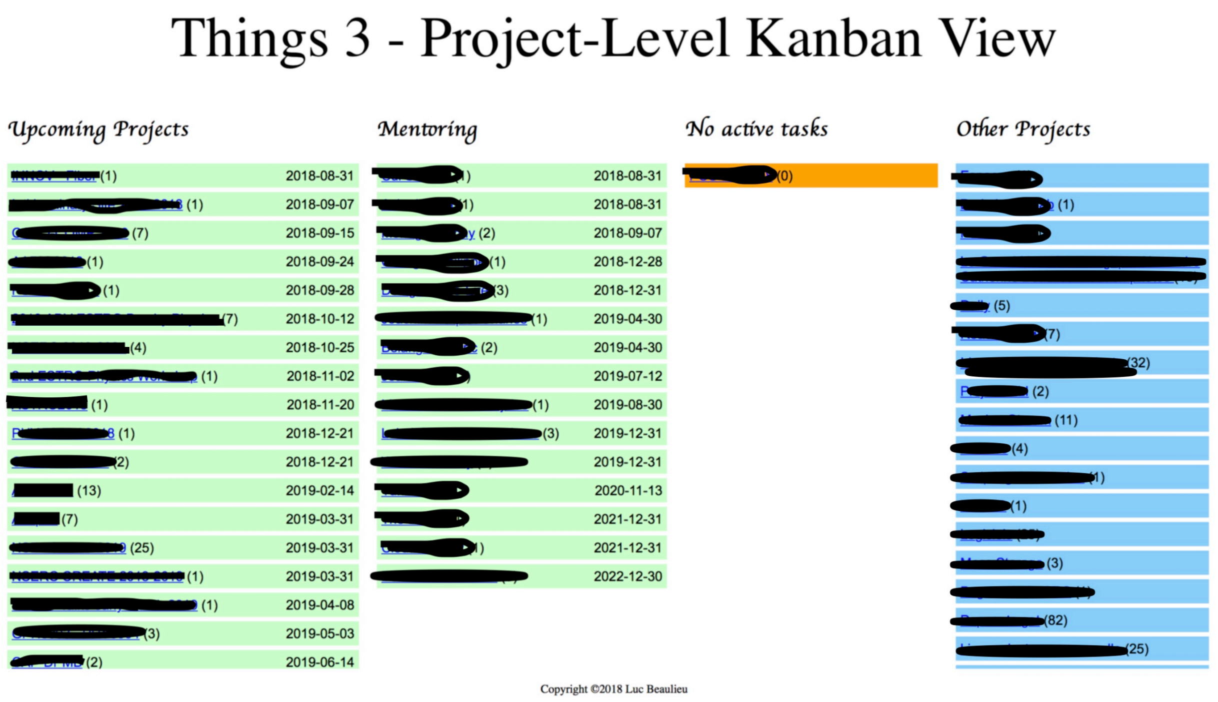
● les champs et listes seulement
○ tous les contrôles

Appuyer sur Contrôle+F7 pour modifier ce réglage.

Configurer un clavier Bluetooth... ?

In the Keyboard system preference panel, chose Shortcut at the top and Services on the left side, go to the bottom where you should see KanbanView (as shown above). Just assign it to an unused combination of keys. You can see that in my case, I already have a bunch of scripts to work with my e-mails, DevonThink and Things.

Whether choosing method 1 or 2, you will see you Kanban display in your browser. In my case, I use a 4-column display. In your case, this vanilla release will produce three columns presented in the past post: Upcoming Projects, No Active Tasks and Other Projects. Note that each column as independent scrolling, which is quite helpful.



Things 3 - Project-Level Kanban View

Upcoming Projects

[redacted](1)	2018-08-31
[redacted] (1)	2018-09-07
[redacted] (7)	2018-09-15
[redacted](1)	2018-09-24
[redacted](1)	2018-09-28
[redacted](7)	2018-10-12
[redacted](4)	2018-10-25
[redacted] (1)	2018-11-02
[redacted] (1)	2018-11-20
[redacted] (1)	2018-12-21
[redacted](2)	2018-12-21
[redacted] (13)	2019-02-14
[redacted](7)	2019-03-31
[redacted] (25)	2019-03-31
[redacted](1)	2019-03-31
[redacted] (1)	2019-04-08
[redacted](3)	2019-05-03
[redacted](2)	2019-06-14

Mentoring

[redacted]1)	2018-08-31
[redacted]1)	2018-08-31
[redacted] (2)	2018-09-07
[redacted](1)	2018-12-28
[redacted](3)	2018-12-31
[redacted](1)	2019-04-30
[redacted] (2)	2019-04-30
[redacted]	2019-07-12
[redacted](1)	2019-08-30
[redacted](3)	2019-12-31
[redacted]	2019-12-31
[redacted]	2020-11-13
[redacted]	2021-12-31
[redacted]1)	2021-12-31
[redacted]	2022-12-30

No active tasks

[redacted](0)

Other Projects

[redacted]

[redacted] (1)

[redacted]

[redacted]

[redacted] (5)

[redacted](7)

[redacted](32)

[redacted] (2)

[redacted] (11)

[redacted](4)

[redacted]1)

[redacted](1)

[redacted]

[redacted] (3)

[redacted]

[redacted](82)

[redacted](25)

Copyright ©2018 Luc Beaulieu

Final note - What would be nice in the future:

- For Cultured Code to implement a Kanban View with various filters ;-)
- For me (or a volunteer) to implement an on-the-fly choice of 3, 4 or n-columns display with user-defined choice of the information contained in each column.

As Cespenar says: enjoy!