main 1 Branch 0 Tags Go to file Go to file Code

bjesus 361f352 · 2 days ago 27 Commits

wren	add basic HTT…	2 days ago
.gitignore	packaging	2 days ago
LICENSE	add license	2 days ago
READM…	add basic HTT…	2 days ago
pyprojec…	packaging	2 days ago
requirem…	update deps	3 days ago
setup.cfg	add license	2 days ago

README MIT license

Wren

a note taking application and a to-do management system that is ridiculously simple, yet very advanced.

About

The simplest task management system with the most advanced features

#cli #todo #telegram #notes

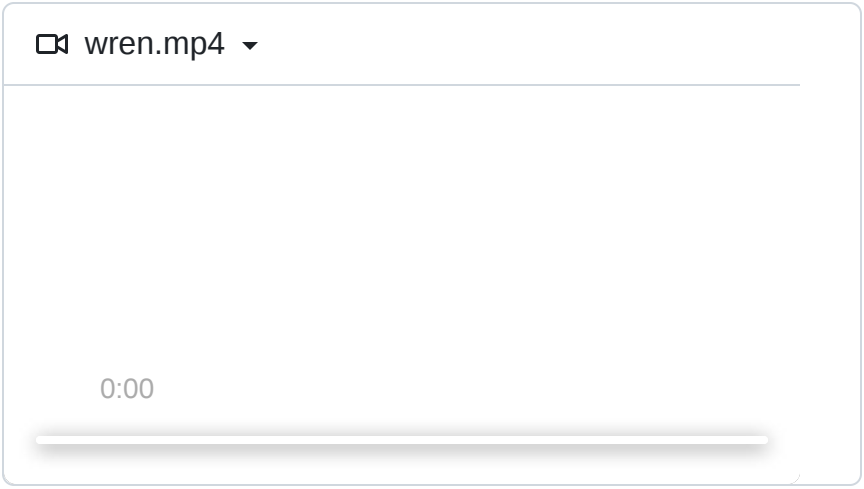
Readme
MIT license
Activity
91 stars
1 watching
0 forks
Report repository

Languages

Python 100.0%

Wren is simple because every note is one file. The filename is the title, and the content is the note's content. This makes it very easy to sync tasks between devices, as conflicts can almost never happen, even if syncing isn't done real time. The files are plain text, so you can just write. If you want a task to repeat every Saturday you just prefix it with a cron syntax, e.g. `0 8 * * 6 weekly swim`, and if you want a task to appear from a specific time you just start it with the date, like `2030-01-01 check if Wren became super popular`.

Wren is advanced because it is very extensible - it comes (optionally!) with a Telegram bot that you can chat with to manage your notes, and even get AI-driven daily summaries as if you had a personal assistant. It also includes a tiny HTTP server that you can use to manage tasks using an API or from the browser, which can be used for displaying you tasks elsewhere (e.g. in your e-reader).



Installation

The easiest way to install Wren is with pip:

```
$ pip install wren-notes
```

Usage

The management of tasks in Wren is simple:

- Tasks are just files living your `notes` folder. You might as well create them with `touch task` and edit them with `vim task`.

- Completed tasks are moved to your `done` folder.
- Tasks starting with a YYYY-MM-DD will not appear in the list of tasks before their time arrived.
- Tasks starting with a cron signature will not be moved when completed. Instead they'll be copied to the `done` directory, and will reappear automatically when the copied file is old enough.

Command line

The regular usage mode Wren is the command line. For the following examples, `n` is my alias to `wren`, but you can use any alias or just call `wren` directly. Normal tasks can be created by just typing them

```
$ n build a spaceship
created task: build a spaceship

$ n go to the moon
created task: go to the moon

$ n 'discuss galaxy peace with aliens
tell them that we won't hurt them
and that we can probably reach some
agreement'
created task: discuss galaxy peace with
aliens
```

Reading a task content is done with the `-r` flag:

```
$ n -r galaxy
discuss galaxy peace with aliens
tell them that we won't hurt them
and that we can probably reach some
agreement
```

Note that when referring to a task, you can give Wren any part of the task title.

For listing your current tasks, just run `n`. Or if you want to filter your tasks, you can use `n --ls query`:

```
$ n
➜ discuss galaxy peace with aliens
➜ go to the moon
➜ build a spaceship
```

```
$ n --ls th
➜ discuss galaxy peace with aliens
➜ go to the moon
```

Use `-e` to edit a task in your `$EDITOR` or `-d` to mark it as done:

```
$ n -d moon
marked "go to the moon" as done
```

Integrations

Random task

Use `--one` to print one random task. I'm using it with Waybar to always have one task displayed at the bottom of my screen, like this:

```
"custom/task": {
  "tooltip": true,
  "max-length": 20,
  "interval": 60,
  "exec": "wren --one"
},
```

AI Assistant

Wren can also work like an AI Assistant. If you use `--summary` it will use GPT4 to create a nice human like message telling you what's waiting for you today, and congratulate you for the stuff you have completed recently. You can use it to update `/etc/motd` daily, or through the Telegram bot (below).

Telegram bot

Using `--telegram` will spin up a Telegram bot listener that will respond to your messages and allow you to create tasks, list them, edit them and so on. It will also allow you to set a cron-based schedule for receiving AI Assistant messages. This can be handy if you want to start your day with a message from Wren telling you about your upcoming tasks.

- List tasks using `/list`
- Create task by just writing it, e.g. `make a plan for going back to earth`
- Mark as done with `/done plan`
- See more at `/help`

If you want to run it outside your computer (e.g. so it's always available), I highly recommend using Syncthing to sync your notes.

HTTP Server

With `--http` you get both a simple tiny website that works through the browser, and an API server that accepts and returns JSON. Either browse to `http://localhost:8080` or send requests directly with the proper headers:

- List tasks: `curl http://localhost:8080`
- Create task: `curl http://localhost:8080 -d '{"task": "create HTTP interface"}' -H 'content-type: application/json'`
- Mark as done: `curl http://localhost:8080/content -X DELETE`

The HTTP server can be used to integrate with voice assistants, Home Assistant, Tasker etc. Like with the Telegram bot, if you want to run it outside your computer, I recommend using Syncthing.

Configuration

See the configuration path on your operating system using `--version`.

The schema is as follows and all keys are optional. Remove the comments from your actual file.

```
{
  "notes_dir": "~/Notes",  // This can
absolute or include ~
  "done_dir": "done",      // This can be
relative to the notes dir, or absolute
  "http_user": "",         // Fill this to
enable basic HTTP auth for the HTTP server
  "http_password": "",     // Same as above
```

```
  "openai_token": "",      // Fill this if
you want to use summaries
  "telegram_token": "",    // Fill this if
you want the Telegram bot
  "allowed_telegram_chats": [
    1234564868             // Initiating a
chat will print out the chat ID you should
fill here
  ],
  // Below you can put context to give the
```