

SCAM Manual

jul

Jan 2025



Table des matières

Synopsis	2
Installation	4
Walkthrough : writing the aide	6
Creating a login	6
your first postit entry	9
Attaching a content to an entry.	10
Developping your first comment and setting your book title	10
Checking on your book	13
Rinse and repeat	14
Playing with the help example	15
Liste des liens	16
WTFPL 2.0 Do any thing you want with this book except claim you wrote it	

Synopsis

This is a front end to a pandoc toolchain based on mind mapping for structuring the thoughts with a real time rendering of the markdown.

Markdown that includes pandoc-include and my own one to integrate graphviz like this :

The home page of SCAM is on [github](#)

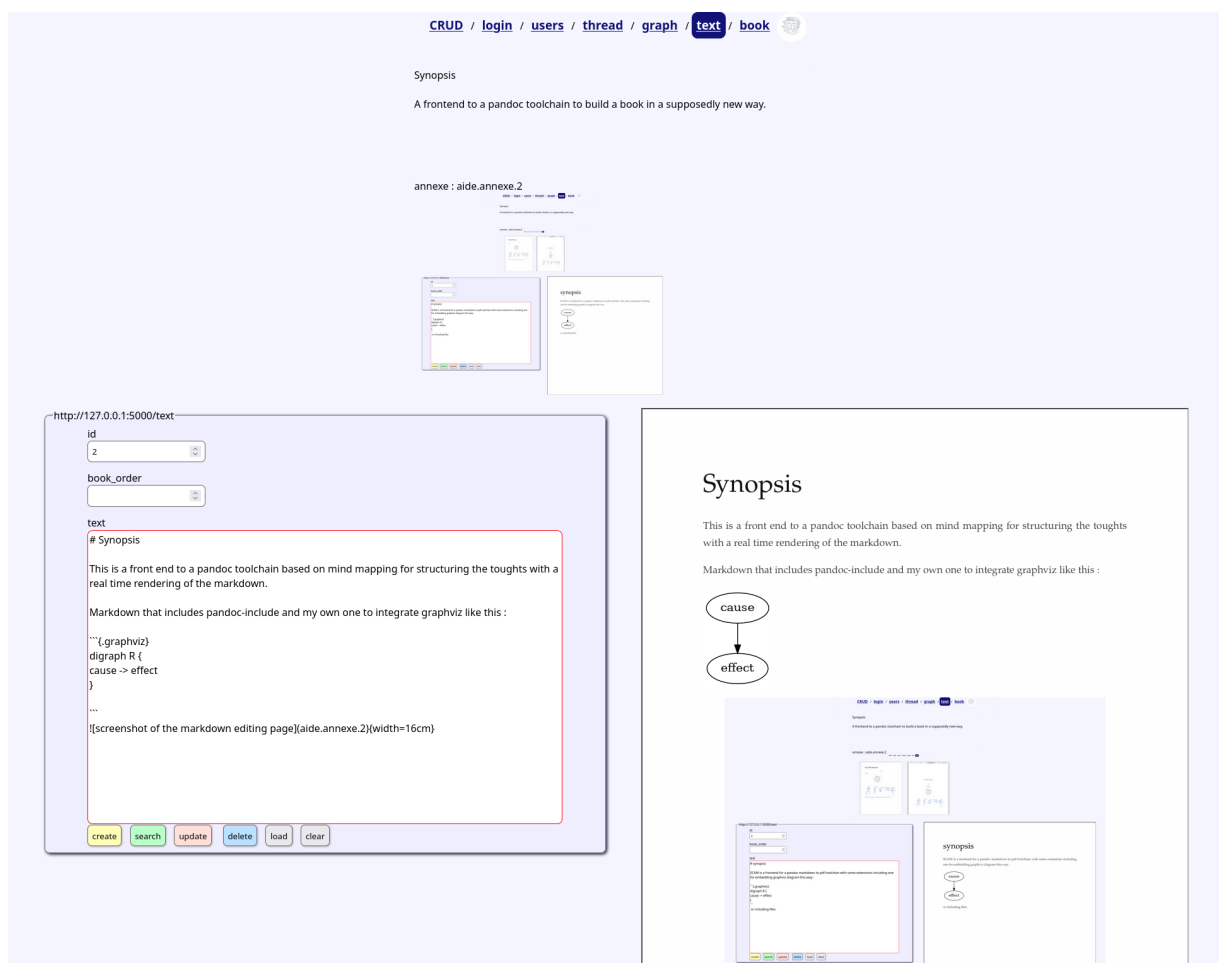


FIGURE 1 – screenshot of the markdown editing page

Installation

I am too lazy to write an install script, make a debian package, or a full pip requirements solution because I need binary installs.

As a result I resorted to the solution of the lazy man which source tells you all you need : making it a docker file.

```
FROM debian
ENV LANG C.UTF-8
RUN mkdir -p /usr/share/man/man1 && mkdir -p /usr/share/man/man7
RUN apt-get update && apt-get -y dist-upgrade \
    && rm -rf /var/lib/apt/lists/*
RUN apt-get update && apt-get -y --no-install-recommends install \
    python3 python3-pip python3-venv python3-setuptools python3-sqlalchemy texlive-
    texlive-fonts-recommended texlive-lang-french lmodern

RUN useradd scam -d /app --uid 1000 -m -s /bin/bash
COPY --chown=scam . /app
WORKDIR /scam
RUN chown -R scam:scam .
RUN mkdir /scam/assets /venv
RUN chown -R scam:scam /venv /scam/assets

COPY . .

RUN virtualenv --system-site-packages /venv
RUN . /venv/bin/activate
COPY requirements.full.txt .
ENV PYTHONPATH=/venv/bin
RUN /venv/bin/python -m pip install --no-cache-dir --disable-pip-version-check -
EXPOSE 5000
CMD . /venv/bin/activate && cd /scam && DB=${db:-scam} /venv/bin/python /app/sca
```

with the following requirements :

```
archery
pandoc-include
dateutils
multipart
filelock
Mako
pandocfilters
panflute
passlib
python-dateutil
SQLAlchemy>=2
SQLAlchemy-Utils
time-uuid
```

to use it I recommend the side car technique wich can be used this way so that you can access the assets dir which contain the book :

```
docker build -t scam .
docker run -i -t -e db=bookname \
    --mount type=bind,src=.,dst=/scam \
    -p5000:5000 scam
firefox http://127.0.0.1:5000
```

Walkthrough : writing the aide

Creating a login

The landing page is a Create Retrieve Update Delete skeleton generated from the data model (which is the HTML of the page by design).

It is recommended to create a user first because all other interesting page are accessible only when connected.

title
id
text

text
id
user id

[CRUD](#) / [login](#) / [users](#) / [thread](#) / [graph](#) / [text](#) / [book](#)

transition
id
previous_comment_id
next_comment_id

comment
created_at_time
id
user_id
comment_id
message
factoid
category

user
id
pic_file
name
email
secret_token
secret_password

annexe
id
annexe_file

http://127.0.0.1:5000/user

id
1

pic_file

Parcourir... cats.png

name

jul

email

j@j.com

secret_token

secret_password

....

create search update delete load clear

FIGURE 2 – CRUD landing page : time to create your user

On successful login you should see this :

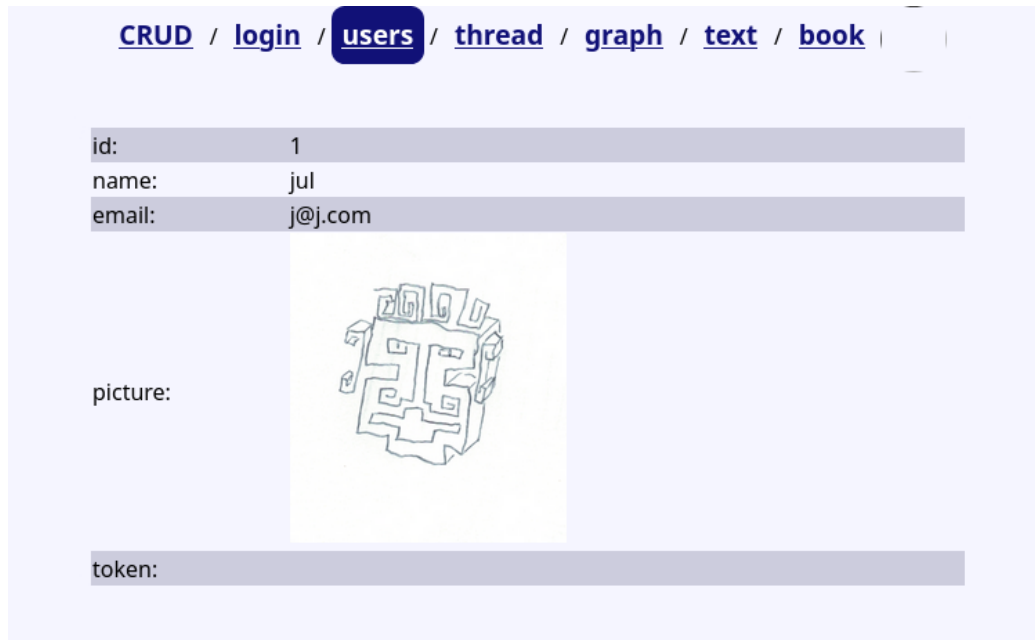
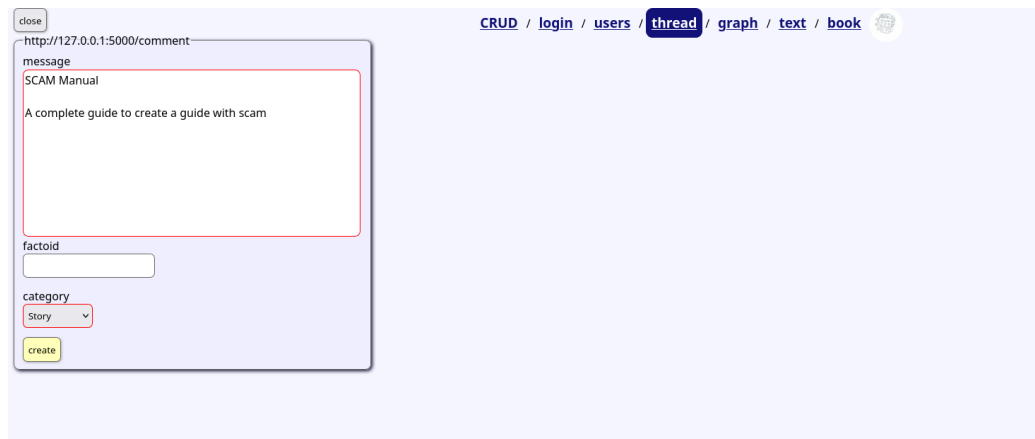


FIGURE 3 – succesfull login

your first postit entry

Time to create your first « idea » ; a small entry of less than 500 characters, like a microblogging entry to know what topic you are gonna talk of



The screenshot shows a web interface for creating a post. At the top, there is a navigation bar with links: [CRUD](#), [login](#), [users](#), [thread](#) (highlighted), [graph](#), [text](#), and [book](#). Below the navigation bar, there is a form titled "message" with the following fields:

- A text input field containing "SCAM Manual".
- A text input field containing "A complete guide to create a guide with scam".
- A text input field labeled "factoid".
- A dropdown menu labeled "category" with "Story" selected.
- A yellow "create" button.

The form is enclosed in a light blue border. The background of the page is light purple.

FIGURE 4 – begin with the **thread** interface



FIGURE 5 – when you correctly add your first entry you have the following output

Attaching a content to an entry.

I chose the policy that one micro item is related to one and only one attachment of the embedable kind you want.

These items will be available in the markdown editor

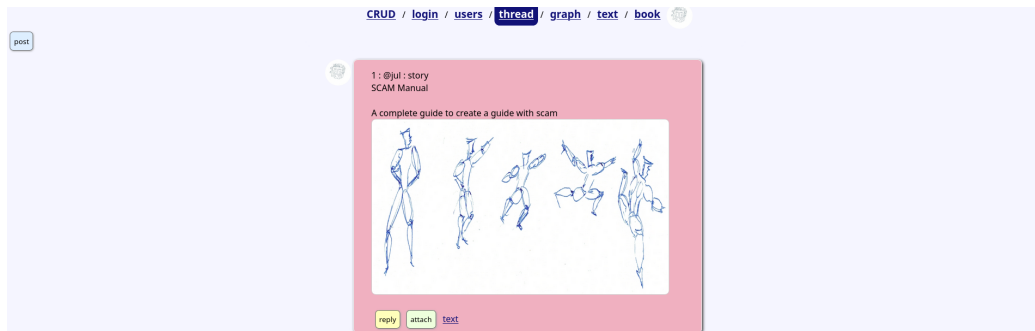


FIGURE 6 – by clicking on the « attach » button you can attach a content to the post it entry

Develloping your first comment and setting your book title

First comment is specific in the sense it is also used for the title. With pandoc you can add metada used for LaTeX.

The *markdown* extension useds here is the pandoc one

Here is the one that I put in the capture

```
% TITLE
% AUTHOR
% DATE \
\
```

```

![] (aide.user.1){width=5cm}
\
\ ![] (aide.annexe.1){width=15cm}

---
header-includes:
- \usepackage[french]{babel}
- \usepackage{hyperref}
- \definecolor{myblue}{rgb}{0.28, 0.24, 0.48}
- \hypersetup{colorlinks=true, allcolors=myblue}
- \let\tmp\oddsidemargin
- \let\oddsidemargin\evensidemargin
- \let\evensidemargin\tmp
- \reversemarginpar
---

```

Your real time markdown input is defintily confused, it is time to add more postists dans develop them.

As you can see text input always have a reminder of the available for inclusion attachment and make it easy to insert it as a picture.

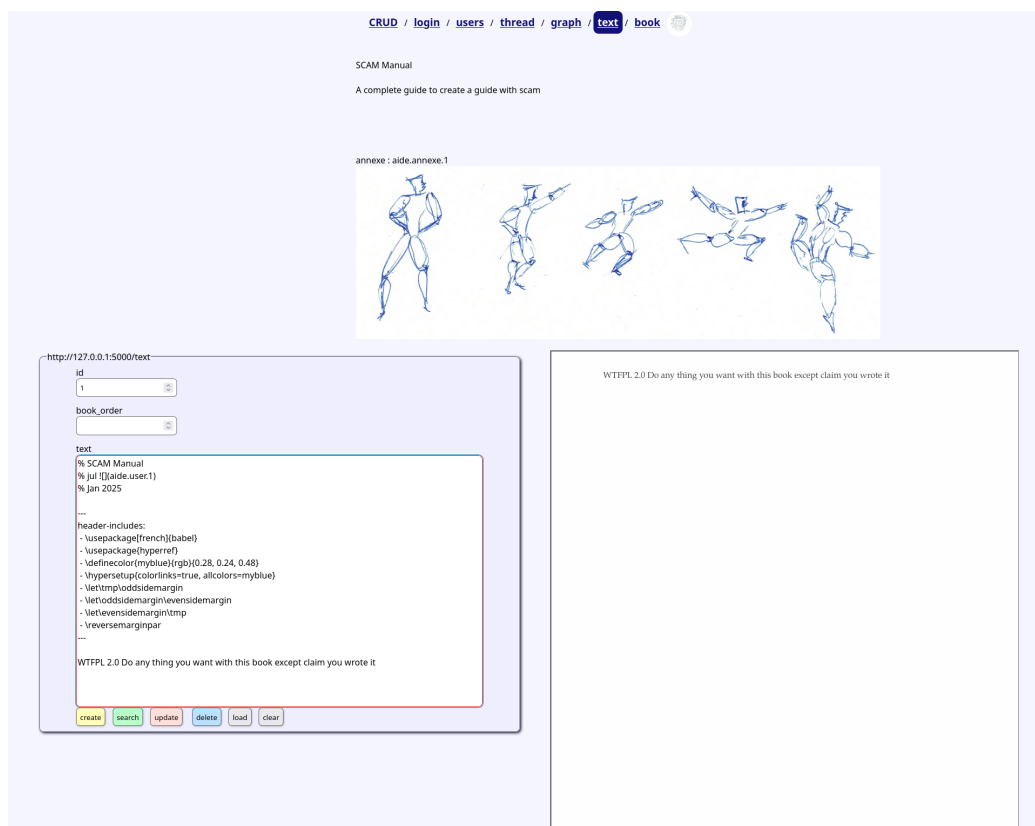


FIGURE 7 – screenshot of the development UI

Checking on your book

it's now time to visit the book rendering URL. You should have a side by side view of the generated standalone HTML and the generated PDF.



FIGURE 8 – book rendering

Rinse and repeat

After a few more entries that are boring because very repetitive if you consult the graph URL you should have now more entries.

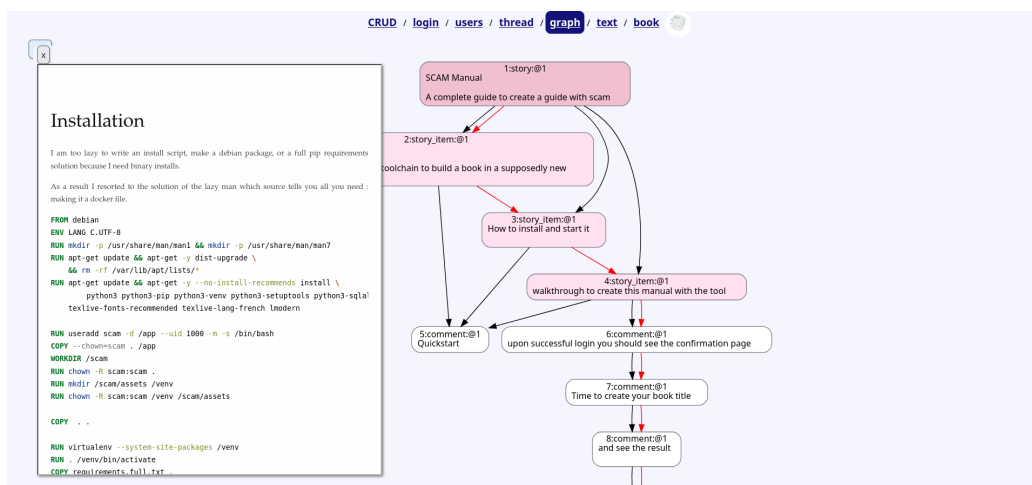


FIGURE 9 – Your graph should now expand itself as the book

The « book order » is the red lines, they follow the ascending id order but can be overridden with the `book_order` rank available in the **text** view.

Playing with the help example

This book is available in the repository as a sqlite database.

To try it :

```
docker run -i -t -e db=aide --mount type=bind,src=.,dst=/scam -p5000:5000 scam
firefox http://127.0.0.1:5000
```


Liste des liens

`http ://127.0.0.1 :5000/` The landing page
`https ://pandoc.org/MANUAL.html#pandocs-markdown`
markdown extension useds here is the pandoc one
`http ://127.0.0.1 :5000/book` it's now time to visit the book rendering URL
`http ://127.0.0.1 :5000/svg` you consult the graph URL
`http ://github.com/jul/scam` The home page of SCAM is on github