

ternbook

User's manual

ternbook a web directory random map

search... try: tag:blog OR title:word

search

tag:a,b

 match tags

title:word

 match name

desc:word

 match description

is:verified

 system tag

-tag:nsfw

 exclude

CORE CONTENT

VIBE / WEB CULTURE

CONTENT SAFETY

TERNBOOK NATIVE

All

All

All

All

reset all

- end of directory -

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Page 3: Information

Page 4: Problems with traditional
directories and what Ternbook offers

Page 6: How-to: Join

Page 6: Part one: Getting the resources
Page 7: Part two: Editing ternbook.json
Page 8: Part three: submitting websites
and IAL codes

How to use this manual

To assist in the use of this Manual the section title is given at the top and the relevant sub-section is given at the bottom of each page.

The purpose of this manual is to assist users, technical and non-technical alike, with usage of Ternbook.

This manual contains explanations of Ternbook's features and procedures for how to submit a website.

Quick Intro of ternbook

Ternbook is a free-to-use, open source, semi-automated, and an easy to join web directory and webring protocol.

Users can submit their websites via either a Web based application or an easy-to-use CLI based program.

Troubleshooting

If you find any troubles regarding usage of Ternbook, or any ternbook.json errors, feel free to ask us or the community about it on <https://github.com/11bunno11/ternbook/discussions/categories/troubleshooting>

Problems with traditional directories

A traditional directory is mainly operated by humans, and half of the time, that alone.

This raises a few problems.

This ranges from tired administrators to link rot, and sometimes directories being "boring" to look at and join into.

Eepy admins!

Administrators have to constantly add, delete, and maintain the directory's infrastructure.

Link Rot!

If a website owner has decided to step away from their website, forget to pay their domain fee, or something like "accidentally broke their backend", this results in the link not becoming active.

A bit boring to look at!

A bit hard to join!

Most directories only offer a fixed "list" view. This is mostly seen on major Directories and Search Engines alike, and is quite boring. Add that to the hardness of joining one. Some may be easy, but a human has to look at your website in order to get it approved and be shown. It may take some time.

What does Ternbook offer?

Ternbook, on the other hand is semi-automated. Automated features include website verification, heartbeat tracking, automatic pruning, and webmap generation.

Automatic Pruning!

Websites that don't interact with an instance within three years will be automatically pruned, resulting in significantly reducing link rot. This also allows instance owners to focus more on things they want to do.

Easy to join!

Website owners just need to add a ternbook.json file in their /.well-known/ directory, and then use either a web based application or a tool we made and recommend, tbflinger.

Interactable webmap!

Ternbook features a D3 force directed graph based, opt-out by default webmap. Users have full control over whether they want to appear or not. You can do boioioioing too :D

Acquiring ternbook.json

You may get your ternbook.json at
<https://ternbook.neocities.org>^{*1}

ternbook / info

[Repo of Ternbook](#)

[How to join Ternbook](#)

[Download JSON](#)

[POST request](#)

Within the website, click

“Download JSON”.

A download will shortly begin.

If the file name is not
ternbook.json, change it to one.

The next pages will show what to
change, what to keep, etc.

It is recommended to work on
editing **while reading the manual**.

^{*1}Information website is subject to change

Editing ternbook.json

Once you download ternbook.json, open the file using Notepad (or any tool of your choice).

Add your website name, the URL, and the description.

```
{
→ "name": "website name, 64 chars max",
→ "url": "https://example.com",
→ "description": "your descriptions here, 256 chars max",
  "tags": ["tag 1", "tag 2"],
  "neighbors": [
    "https://example.com"
  ],
→ "map": "out", ← for webmap, opted out by default, if
                  joining, change "out" to "in"/"join"
  "ial": "refer to document for more information"
}
```

Next, input tags. Maximum number of tags is 10*¹.

Below are the default canonical tags Ternbook supports*².

Core Content Tags	vibe/web culture tags	Content safety/Filtering tags
blog	web1	nsfw
portfolio	webcore	political
wiki	y2k	ads
personal	cyberpunk	ai-generated
project	minimal	satire
archive	brutalist	
docs	retro	
forum	terminal	
social	weirdweb	
indie	handmade	
startup	cozy	
tools	chaotic	
search	aesthetic	
media	oldnet	
gallery	lowtech	
journal	experimental	
experiments	glitch	
learning	vaporwave	
education		
gaming		
music		
art		
writing		
photography		
coding		
devlog		
maps		
recipes		
news		
links		
directory		
microblog		

*¹Maximum tag numbers is subject to change.

*²Canonical tags are subject to change.

Editing ternbook.json (continued)

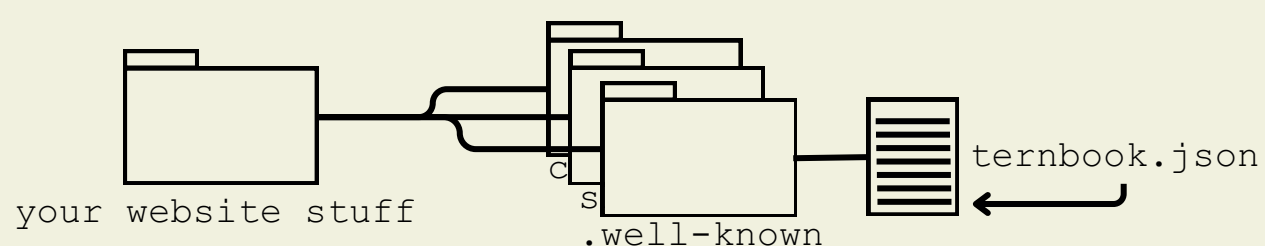
Tags must be inputted in the shape below. Or else, Ternbook cannot detect the tags.

```
Single tag:"tags": ["tag 1"],
Two tags:  "tags": ["tag 1", "tag 2"],
Three tags:"tags": ["tag 1", "tag 2", "tag 3"],
           :
```

The neighbors section must be inputted like this, reason is the same as above.

```
Single neighbor:"neighbors": [
    "https://example.com"
],
Two neighbors:  "neighbors": [
    "https://example.com",
    "https://example2.com"
],
Three neighbors:"neighbors": [
    "https://example.com",
    "https://example2.com",
    "https://example3.com"
],
           :
```

Once you've finished adding the necessary sections create a `/.well-known/` directory within your website, and put `ternbook.json` within that folder.



*¹Maximum tag numbers is subject to change.

*²Canonical tags are subject to change.

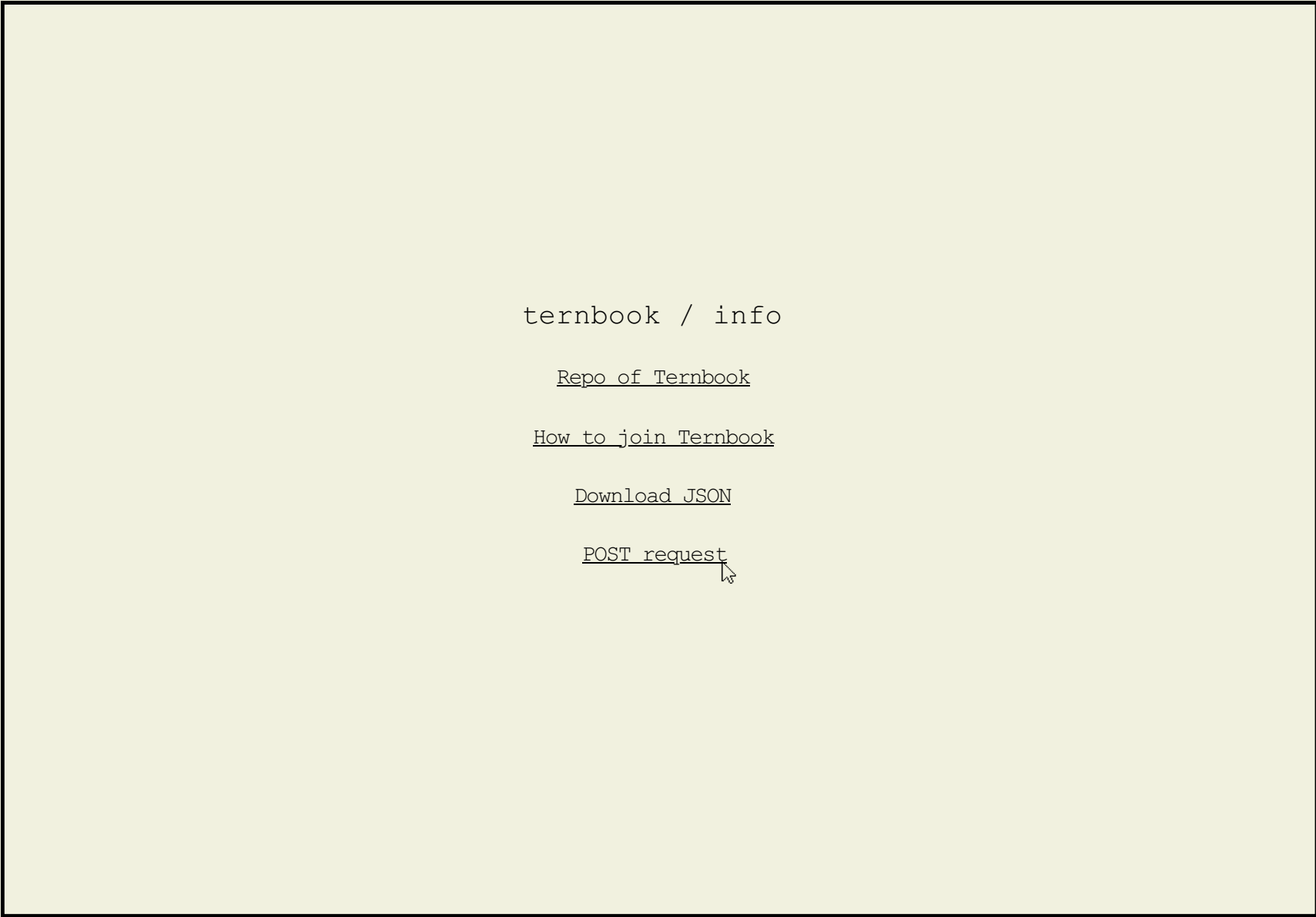
Submitting your website

Once you've entered the necessary sections, you're now ready to submit your website. **Read the manual carefully before continuing.**

You have 2 ways to submit your website. First is the Web application, and the second is the recommended CLI application, `tbflngr`.

Using the web application

You may enter the web application at <https://ternbook.neocities.org>^{*1}

A screenshot of a web browser showing the 'ternbook / info' page. The page has a light blue background with a white rectangular box in the center containing a list of links. The links are: 'Repo of Ternbook', 'How to join Ternbook', 'Download JSON', and 'POST request'. A mouse cursor is hovering over the 'POST request' link.

```
ternbook / info
Repo of Ternbook
How to join Ternbook
Download JSON
POST request
```

Please keep in mind this web application only allows submission to <https://ternbook.onrender.com>^{*2} so anyone who want to submit their website to other instances may use the recommended CLI application.

^{*1}Information website is subject to change

^{*2}"Central hub website" is subject to change

Submitting your website

Using the recommended CLI application

Using pip, you may install Ternbook's CLI tool, **tbflingr**^{*1}.

Within the command prompt (or your preferred CLI tool), enter

pip install tbflingr

if pip only didn't work, enter

python -m pip install tbflingr

Once you install the tbflingr CLI application, you only have to enter your website, specify the instance, and submit. **Do not exit the application yet.**

tbflingr fling example.com to exampleinstance.com (save profile [num])

Full tbflingr commands can shown by typing **tbflingr** only.

After you submit your website, text like this below will show up.

Within the text, there will be something called an "IAL"

(I Am aLive) Code. Copy that code, add that to your "IAL" section, and your website will be verified in the next ping. A ping can be sent every 12 hours.

IAL codes **must be changed every**

2 months. A non verified website's visibility will be restricted to search results only after a week.

```

Fetching https://example.com/.well-known/ternbook.json
Flinging→ https://exampleinstance.com/api/heartbeat
← 200

```

```

Status      OK
URL         https://example.com
IAL         xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Prev IAL    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Epoch      xxxxx
IAL Verified false

```

^{*1}<https://pypi.org/project/tbflingr/>

[illegible]

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

