

---

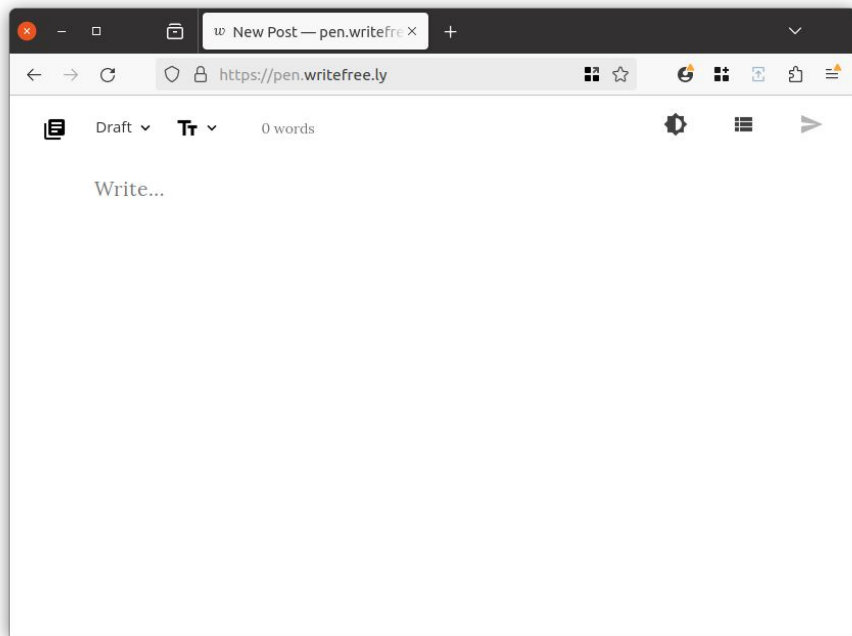
---

# Federated Blogging with WriteFreely

By Matt Baer  
Lead developer, WriteFreely

---

# *write freely*



---

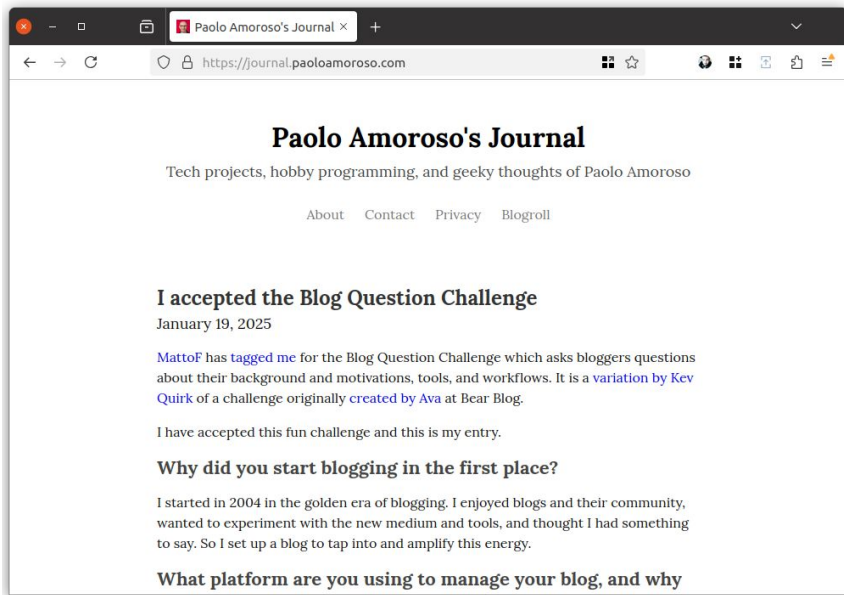
# What is WriteFreely?

- Minimalist blogging platform made just for writing
  - Licensed AGPL
  - Written in Go (golang)
  - Easy to install and maintain (single static binary)
  - Runs on very limited resources
  - Built around plain text / Markdown
  - Web, RSS, email subscriptions, ActivityPub, Gopher
  - Backed by SQLite or MySQL / MariaDB
-

---

# What is WriteFreely?

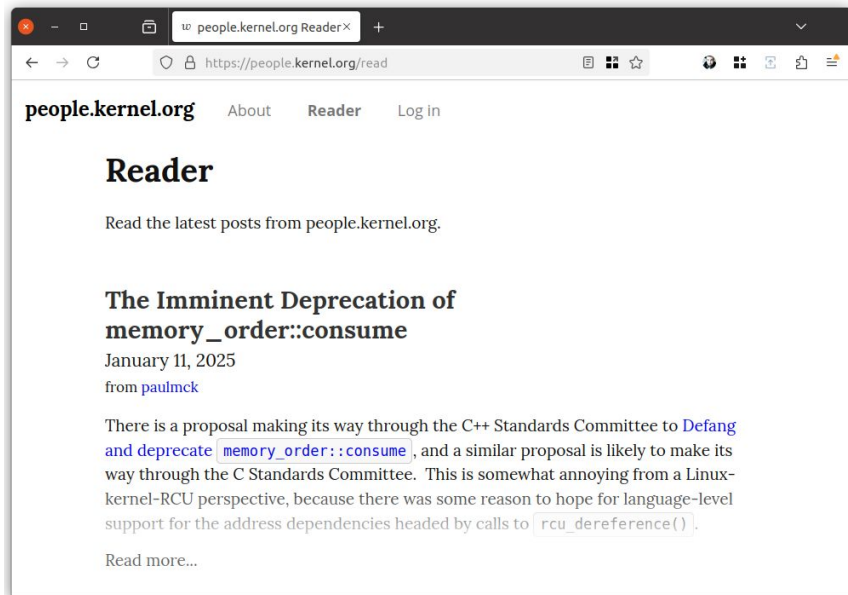
Can be set up for a single user.



---

# What is WriteFreely?

...or configured for a community, organization, etc.



# Set Up

First, [download the latest release](#) for your operating system. If your OS and system architecture are not listed, you'll need to [build from source](#), instead.

If you plan to use **MySQL** for your database, separately connect to your MySQL server now and run:

```
CREATE DATABASE writefreely CHARACTER SET latin1 COLLATE latin1_swedish_
```

Extract the files and go into the project folder. Next, start the configuration process by running the following command. This will walk you through the various ways you can configure your instance:

```
writefreely config start
```

Generate the encryption keys for your instance by running:

```
writefreely keys generate
```

And finally, run:

```
writefreely
```

*from:*  
[https://writefreely.org/  
start](https://writefreely.org/start)





mattbaer@polaris: ~



## 🔧 WriteFreely Configuration 🙌

This quick configuration process will update the application's config file, demo.ini.

It validates your input along the way, so you can be sure any future errors aren't caused by a bad configuration. If you'd rather configure your server manually, instead run: `writefreely --create-config` and edit that file.

### Server setup

Production, standalone  
Insecure (port 80)

### Database setup

SQLite  
Filename: writefreely.db

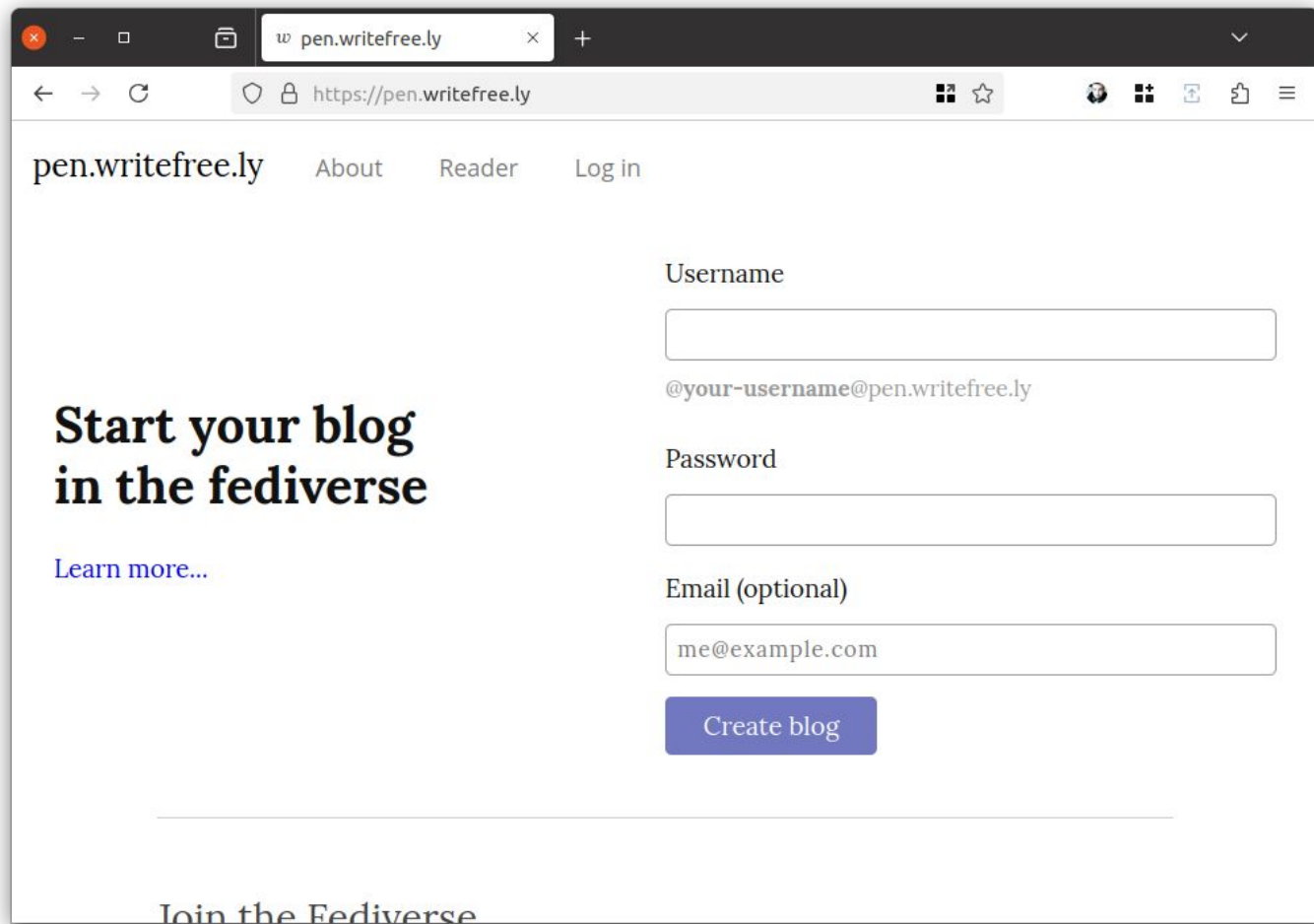
### App setup

Use the arrow keys to navigate: ↓ ↑ → ←

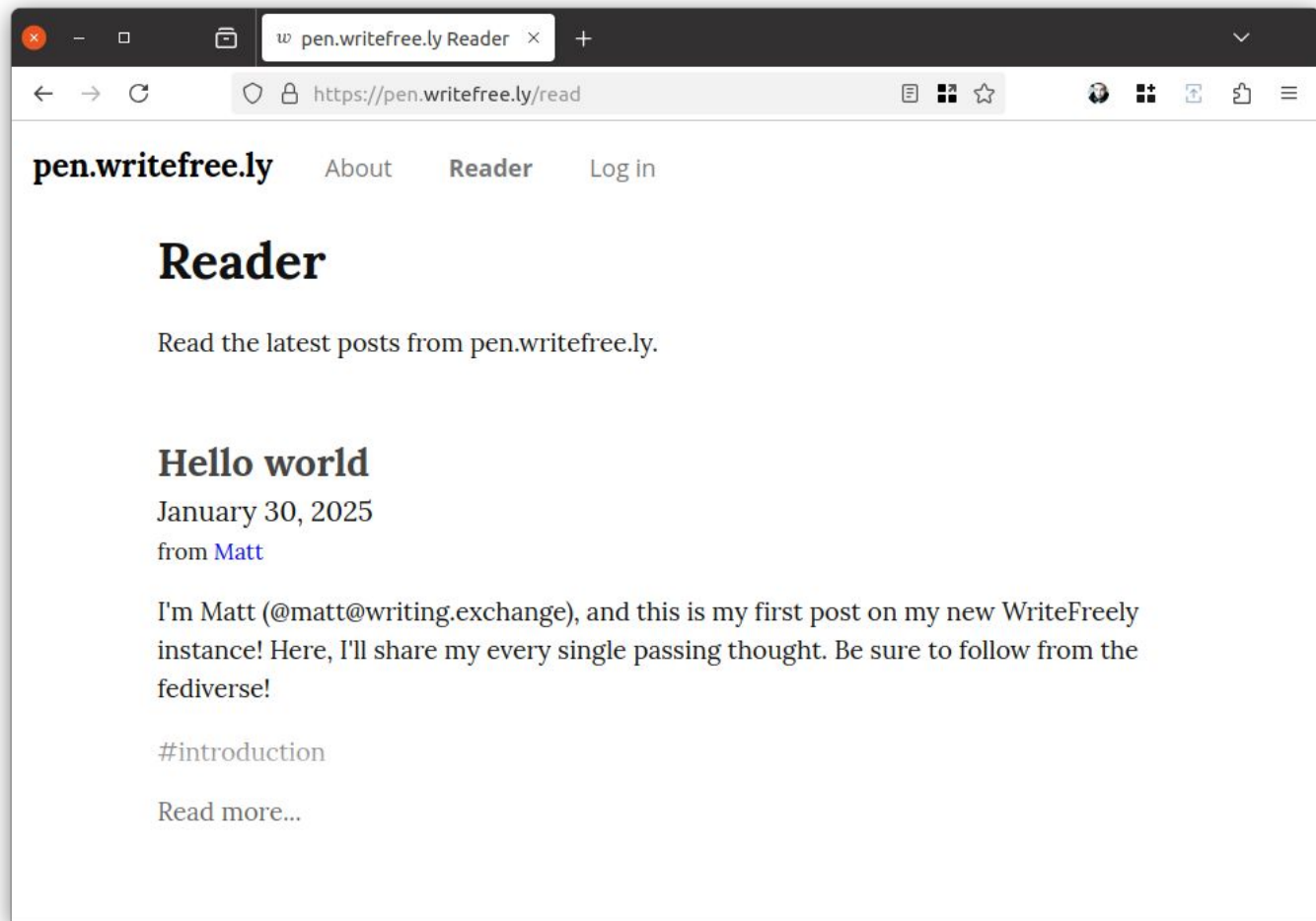
? Site type:

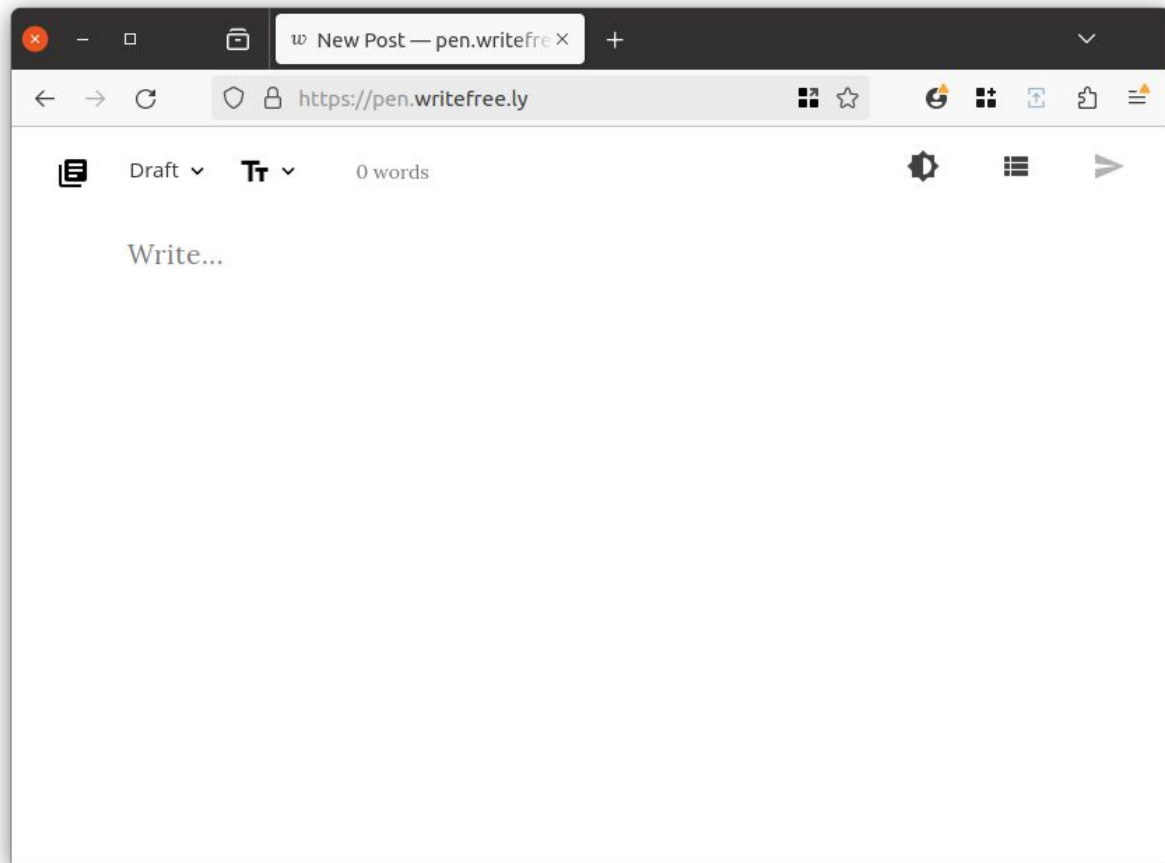
Single user blog

▶ Multi-user instance









---

# Further configuration

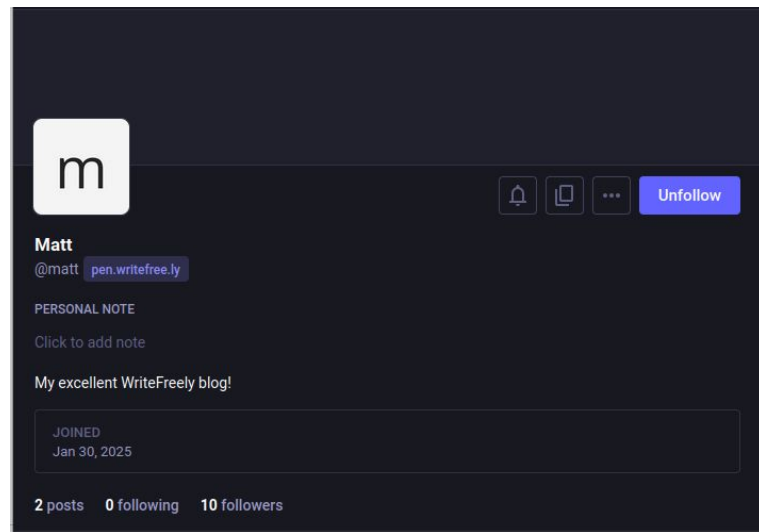
More examples:

- Switch to rich text editor
  - Allow multiple blogs per user
  - Federate each post as a Note instead of Article
  - Bulk email provider (currently only Mailgun)
  - Sign-ups: open, invite-only, closed
  - OAuth provider (Slack, Gitlab, Gitea, generic)
  - Completely private
-

---

# WriteFreely in the fediverse

Anyone on Mastodon, etc. can follow a WriteFreely blog, receive posts in their feed, and interact with it directly.



✕

—

□

📁

Matt: "Hello world https://x

+

▼

←

→

🔄

🛡️

🔒

📶

https://writing.exchange/@matt@pen.writefree.ly/1

Work 📁

🗖️

★

👤

🗑️

🔖

📌

🔔

mastodon

🔍

New post

W

← Back

👁️

m

Matt

@matt@pen.writefree.ly

Hello world

pen.writefree.ly/matt/hello-wo...

m

Matt · 1d

Hello world

I'm Matt (@matt@writing.exchange), and this is my first post on my new WriteFree...

#Introduction

Jan 30, 2025, 08:10 PM · 🌐

0 boosts · 1 favorite

↩️

↺

★

📌

⋮

Replies from other servers may be missing

🏠 Home

🔔 Notifications

🔗 Explore

🌐 Live feeds

👤 Private mentions

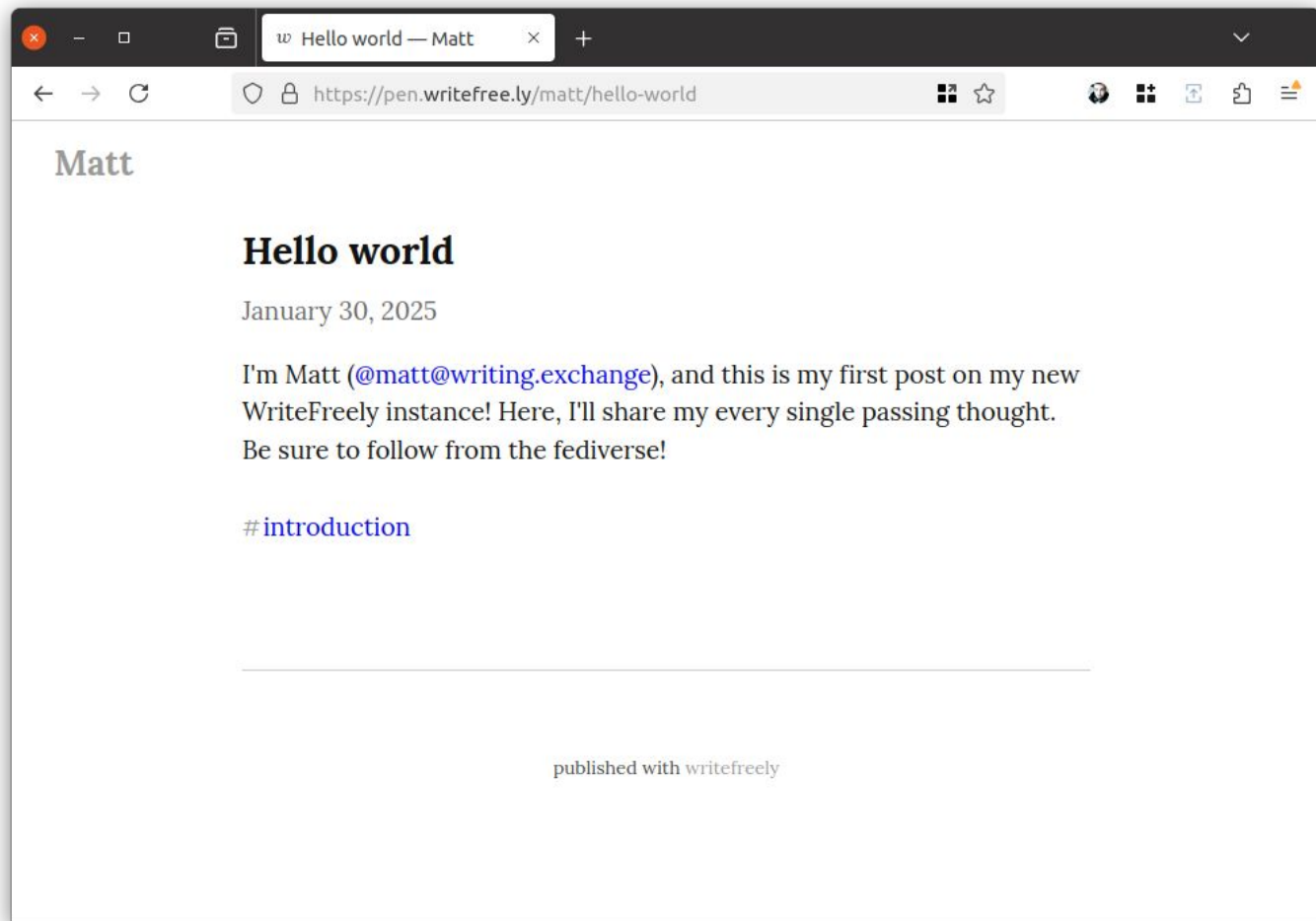
📖 Bookmarks

★ Favorites

📋 Lists

⚙️ Preferences

⋮ About



Matt

## Hello world

January 30, 2025

I'm Matt ([@matt@writing.exchange](mailto:@matt@writing.exchange)), and this is my first post on my new WriteFreely instance! Here, I'll share my every single passing thought. Be sure to follow from the fediverse!

#[introduction](#)

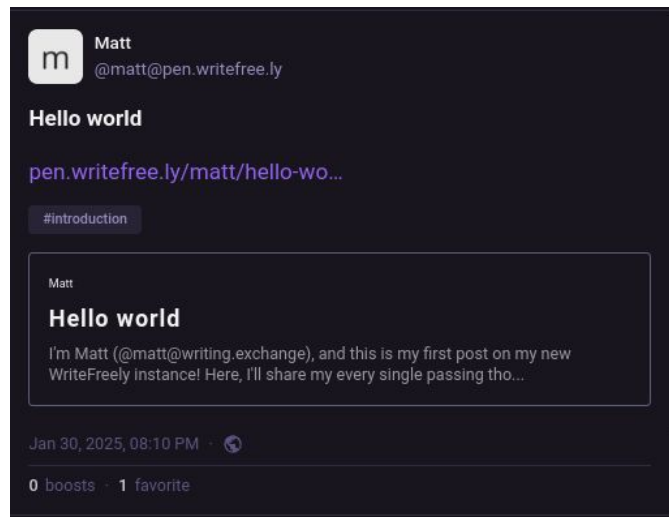
---

published with [writefreely](#)

---

# WriteFreely in the fediverse

Hashtagged posts show up in searches on other platforms



---

# WriteFreely in the fediverse

- Images show up on other platforms
  - Mention other fediverse users from a WriteFreely post (e.g. @matt@writing.exchange) – it'll link to their profile and they'll be notified
  - Readers can *like* posts, and author will see a total number of likes on WriteFreely
  - Readers can *boost* blog posts if a platform allows, though the author isn't notified on WriteFreely
-



—

Joining the fediverse

---

# write.as

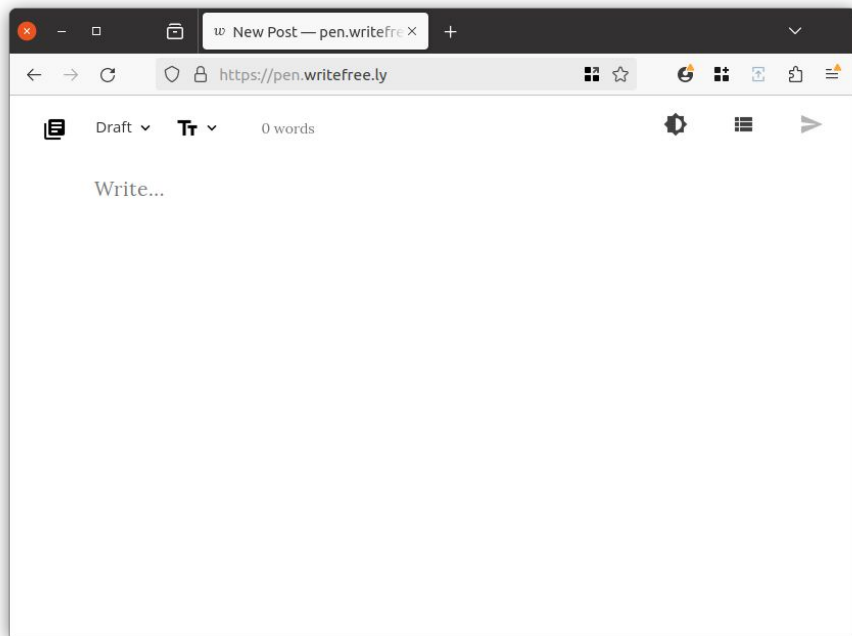
- Launched in 2015
  - Focused on simplicity, privacy
  - Foundation for WriteFreely
  - Now our flagship instance
-

—

# Implemented ActivityPub in 2018

---

# *write freely*



—

...it was a little  
painful

---

# Implementing ActivityPub

- Little documentation outside of the spec
  - Properties can be any type (string, object, array)
  - Most platforms conform to how Mastodon implements ActivityPub
  - Few pre-built libraries
-

# Used [go-fed](#) library



Version: v0.4.0

[Go to latest](#)

| Published: Nov 17, 2018

| License: [BSD-3-Clause](#)

| Imports: 4

| Imported by: 34

## README

What it does

Using concrete types

Serializing data

What it doesn't do

Other considerations

Documentation

Source Files

## Index

### type Accept

```
func NewAccept() (n *Accept)
func (t *Accept) AppendActor(k *url.URL)
func (t *Accept) AppendAttachment(i vocab.ObjectType)
func (t *Accept) AppendAttachmentLink(i vocab.LinkType)
func (t *Accept) AppendAttributedTo(k *url.URL)
func (t *Accept) AppendAudience(k *url.URL)
func (t *Accept) AppendBcc(k *url.URL)
func (t *Accept) AppendBto(k *url.URL)
func (t *Accept) AppendCc(k *url.URL)
func (t *Accept) AppendContent(k string)
func (t *Accept) AppendContext(i vocab.ObjectType)
func (t *Accept) AppendContextLink(i vocab.LinkType)
func (t *Accept) AppendGenerator(i vocab.ObjectType)
func (t *Accept) AppendGeneratorLink(i vocab.LinkType)
func (t *Accept) AppendIcon(i vocab.ImageType)
func (t *Accept) AppendImage(i vocab.ImageType)
func (t *Accept) AppendInReplyTo(k *url.URL)
func (t *Accept) AppendInstrument(i vocab.ObjectType)
func (t *Accept) AppendInstrumentLink(i vocab.LinkType)
func (t *Accept) AppendLocation(i vocab.ObjectType)
func (t *Accept) AppendLocationLink(i vocab.LinkType)
func (t *Accept) AppendName(k string)
func (t *Accept) AppendObject(i vocab.ObjectType)
func (t *Accept) AppendOrigin(i vocab.ObjectType)
func (t *Accept) AppendOriginLink(i vocab.LinkType)
func (t *Accept) AppendPreview(i vocab.ObjectType)
```

# A [blog post](#) on how to do the basics

[← Back](#) Jul 3, 2018 [Guides](#)

## How to make friends and verify requests

Implementing an ActivityPub inbox



Eugen Rochko  
CEO / Founder

In the previous tutorial we have learned [how to send a reply to another ActivityPub server](#), and we have used mostly static parts to do it. Now it's time to talk about how to subscribe to other people and receive messages.

### The inbox

Primarily this means having a publicly accessible inbox and validating HTTP signatures. Once that works, everything else is just semantics. Let's use a Sinatra web server to implement the inbox.

In fact, I intend to omit persistence from this tutorial. How you would want to store data in a real application is very much up for debate and depends on your goals and requirements. So, we're going to store data in a variable and implement a simple way to inspect it.

```
require 'sinatra'

INBOX = []

get '/inspect' do
  [200, INBOX.join("\n\n")]
end
```



—

it's not so simple...

---

# write.as

- Hosts 500k+ users and 5.5 million+ posts
- Subset of federated blogs have 10k+ followers collectively
- ...but they've still managed to take down the entire platform

---

# Growing pains

Fediverse accounts with large followings would cause downtime and dropped requests when boosting WriteFreely posts.

**Solution:** caching on commonly-requested endpoints (actor, inbox, post)

---

---

# Growing pains

Blog followers were on instances with different uptimes and slow response times.

**Quick solution:** slightly longer timeouts, parallel requests instead of serial

**Long-term solution:** implement a queue for making requests, with retries

---

---

---

# Growing pains

New platforms join the fediverse with varying levels of compatibility.

**Solution:** file a bug report

---

—

What's next?

---

# What's next?

Full two-way communication with the fediverse!

- Can accept replies
  - Moderation and safety is huge
  - Hope to support short and long-form comments from WriteFreely
-

---

# What's next?

Better long-form Article support?

- Attempted with Read.as in 2018
  - [Hometown](#) (Mastodon fork)
  - Long-form readers: Ghost, Flipboard
  - Support from major fediverse platforms
-



---

---

What's next?

Slow, steady improvements

---

---

---

# Want to help?

- Join our forum: [discuss.write.as](https://discuss.write.as)
  - Contribute: [writefreely.org/contribute](https://writefreely.org/contribute)
  - Roadmap: [writefreely.org/tasks](https://writefreely.org/tasks)
-

---

# Thank you!

**Matt Baer**

m: [@matt@writing.exchange](mailto:@matt@writing.exchange)

e: [matt@write.as](mailto:matt@write.as)

**WriteFreely**

<https://writefreely.org>

**Write.as**

<https://write.as>

---