

Etherchat

Adventures in Irritation Driven Development
h/t to Eddie Lay

Making a web interface to the ChatGPT API is so common now that it's almost the "hello world" of AI applications.

I want to do something different with this project.

I have over 500 logins stored in IPassword. Only maybe twenty of them are for banks or paid services where a user account is actually necessary.

In many cases I simply wanted to look at something and then never visited that website again, leaving behind a useless account linked to my email.

This project questions this design assumption and explores an alternative approach.

What features do users expect from user accounts?

- privacy - my chats are my own
- sharing - sharing is my choice

How are these features provided by existing services without user accounts?

- privacy - <https://yakgpt.vercel.app> uses a uuid to generate an anonymous page
- sharing - <https://etherpad.org> uses a crdt data structure to share anonymous pages

How can we provide these features in Phoenix LiveView?

- privacy - random string generator for new pages
- sharing - Phoenix PubSub

Controller code to generate a random page

```
def home(conn, _params) do
  chat = MnemonicSlugs.generate_slug(3)
  redirect(conn, to: "#{chat}")
end
```

Then put the live page first in the router

```
live("/:chat", ChatLive)  
get("/", PageController, :home)
```

Code to allow the user to create a new random page

```
push_navigate(socket, to: "/", replace: true)
```

Code to allow the user to create a custom page

```
new_page = URI.encode_www_form(custom)  
push_navigate(socket, to: "{new_page}", replace: true)
```

This project is designed for coders with Phoenix PubSub providing real-time sharing for pair programming.

And having no requirement for user accounts makes using it friction free.

Advantages and disadvantages of Phoenix LiveView

Phoenix PubSub made real-time sharing easy, literally 3 lines of code, where similar apps like Etherpad rely on complicated but perhaps more featureful crdts.

The static front page of an Etherpad site makes it easy for a user to understand usage. The LiveView single page architecture required more attention to UI design.

Project site -> <https://etherchat.net>

Project repo -> github.com/loteks/etherchat

Etherpad demo -> <https://indiepad.net>

We find hot new industry trends & then do the opposite of that...

— Big Sky Software

Many thanks to Brooklin and all my Cohort peeps :)

