

Taking it further

In this section you will

- Do it yourself!

At this point, you've learned everything you need to know about Vue in order to continue developing your chat client by yourself. Here are some challenges:

Create a memberlist component.

- Create a new memberlist component, passing in your chat server.
- Create a new response stream specifically for your memberlist
- Use the stream to populate your component's properties, and bind them to your html
- On a timer, send a memberlist command to the server to get the latest memberlist

Style your app

- Can you use component's .scss files to style your components and application?

Join multiple rooms

- Create a component that contains an input field that you can type room names into
- Bind that input field to your component, and when you submit on your web page, send a join command to the server.
- In your new component, list all the rooms you have currently joined
- When you click a room in your list, make that room's chat appear in your chat window.
- Can you refactor your solution to preserve message history when switching between rooms?

There is a lot to learn about Vue, and what we have created is the tip of the iceberg. There are many ways to interact and code in Vue, some of which would be considered a 'happier path' for our chat server tonight. Hopefully you have gained an understanding of some of the underlying features and wish to research this further.

One of the major keys to success in any reactive front-end framework like Vue is how you create your components. It could be worth looking into configurable components, slots, dynamically creating components, layouts, and event systems. I hope you have enjoyed exploring this chat client and thank you for participating.