

# 聊天網站實作

原始碼 <https://github.com/maxhu787/react-redis-chat>

## 使用語言與框架



- **前端:** ReactJS, Sass, Chakra UI
- **後端:** Node.js, Express, Socket.io, Redis, PostgreSQL
- **其它:** framer-motion (動畫)、Formik (表單驗證)、npm、git、Docker

## 主要功能

- **創建帳號**、個人檔案 (頭像、自介等)、加好友、**創建群主**
- **即時訊息**、**上線狀態**、login session (可以不用每次重登入)
- **傳送圖片**、表情符號、image embedding (圖片嵌入)

## 實作

- **帳號**：用戶帳號資料存儲於 postgres 資料庫，包括用戶 ID、用戶名、密碼 hash、頭像和個人簡介。用戶登錄時，login session 會存到 redis，這樣一個禮拜內就不用再重新登錄，我用 express-session + redis。
- **訊息**：訊息發送時通過 socket.io，並即時廣播給其他連接的用戶，訊息同時被後端 server 存到 redis 中，方便快速讀取。
- **群主、好友**：這些關係性的資料也選擇存儲在 redis 中以提高性能，postgres 則是僅用於儲存帳號的資料。
- **前端**：前端使用 Chakra UI，並利用 Framer Motion 做動畫效果 (載入畫面 logo、新訊息彈出的動畫等)。
- **其他**：使用 Docker 容器，系統相容性

# CHAT APP

---

310125 胡凱翔

---

# 動機

剛開學時，Instagram帳號無緣無故被封鎖，寄email到客服也救不回來，所以就想要做自己的一個聊天網站



# 期望成果

可以建立帳號、加朋友聊天、傳送emoji、圖片

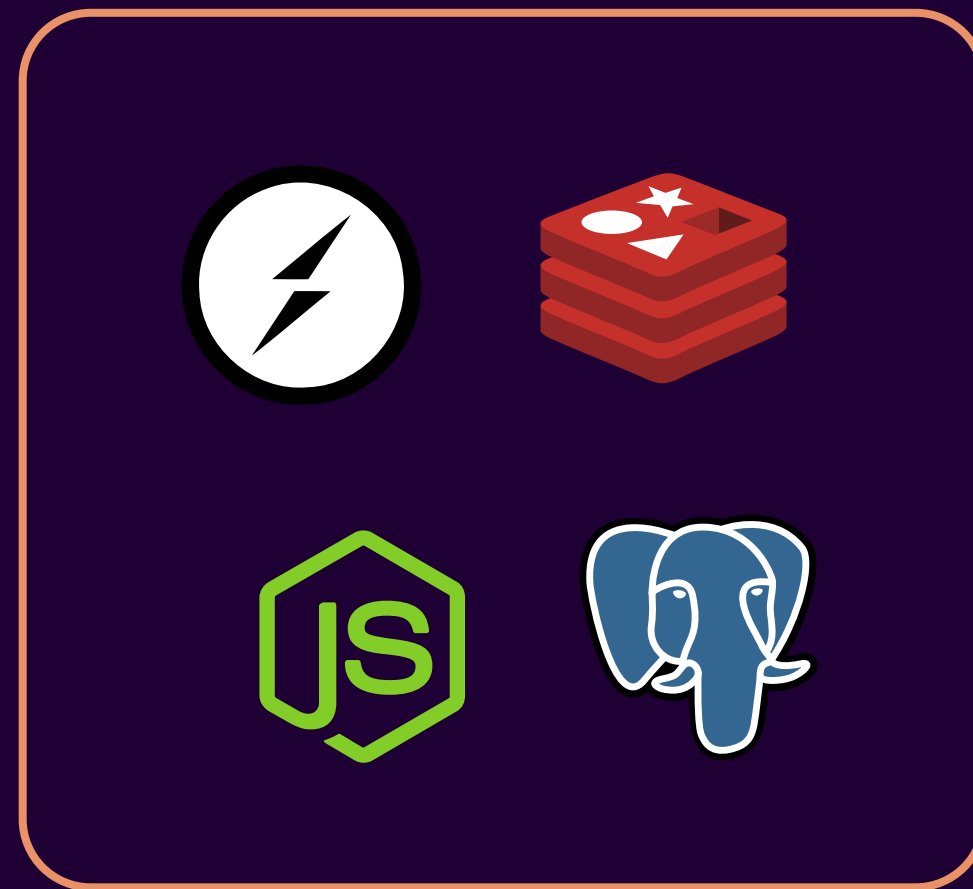
# TECH STACK

---

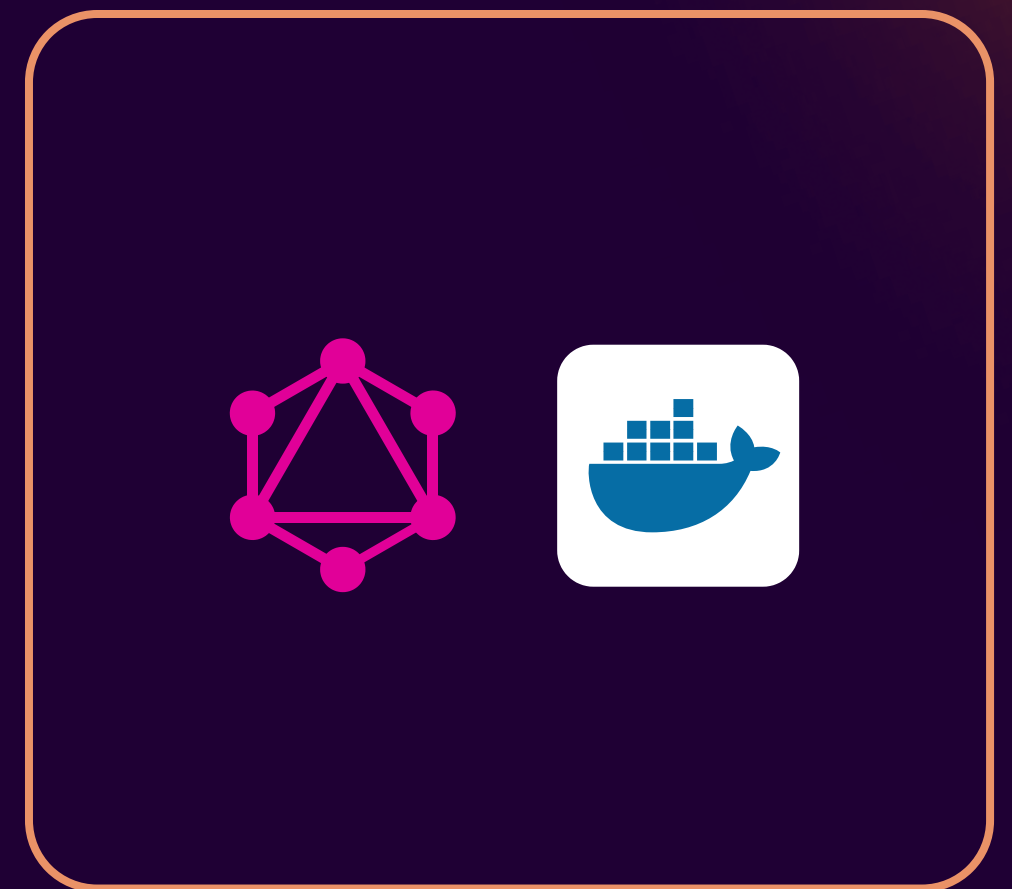
## Frontend



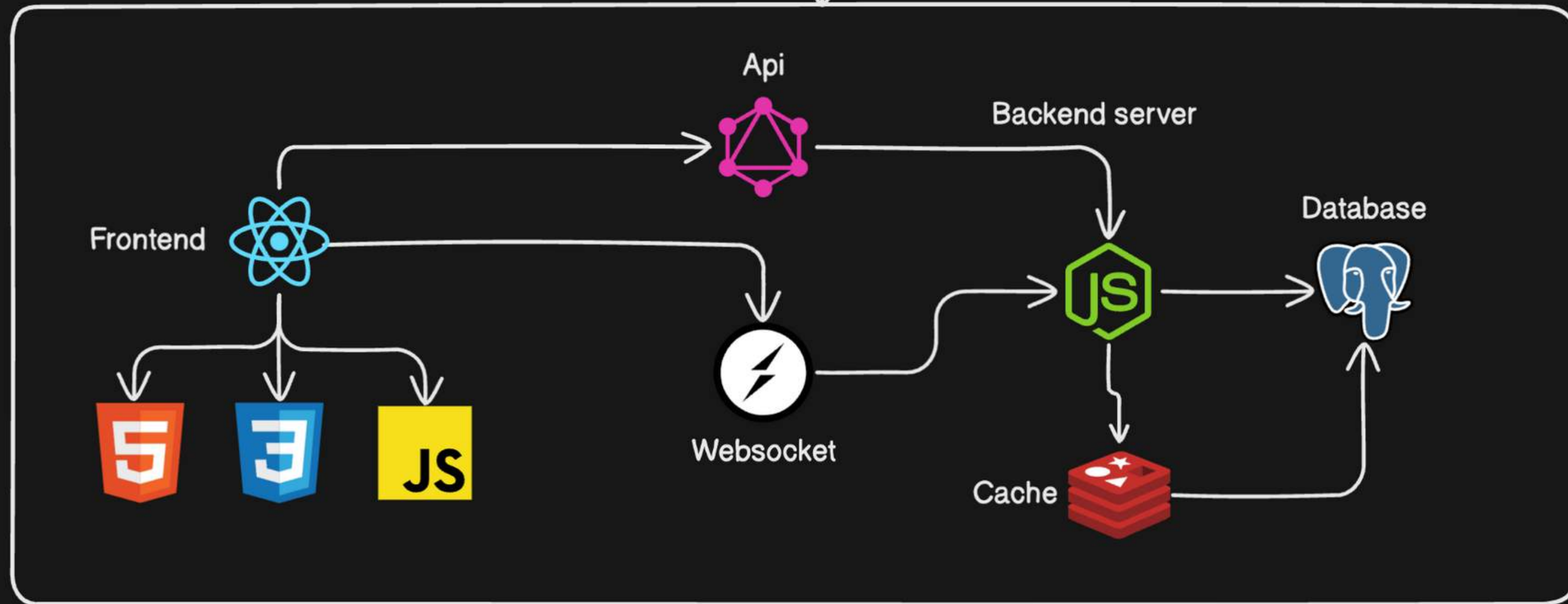
## Backend



## Other stuff



Docker container



TCSN • test



maxmax

You: 😊 7

No groups found :(



hukaixiang



Chat



fefo qwifeu huiwhef ouhqwef iuhqwef

qwef

wqefiuh

euhi jdhf ee nbf mnae iuh

wqefi qwuehfiuqwe

qwefi

qwefiwqeufh

qwefiuh

qwef

qwieufhqef



qwef



Type in your message...



GIF

Send

TCSN • test



maxmax

You: 😊 7

No groups found :(



hukaixiang



Chat



fefo qwifeu huiwhef ouhqwef iuhqwef

qwef

wqefiuh

euhi jdhf ee nbf mnae iuh

wqefi qwuehfiuqwe

qwefi

qwefiwqeufh

qwefiuh

qwef

qwieufhqef



qwef



Type in your message...



GIF

Send



TCSN • test



Chat



maxmax

You: 😊 7

No groups found :(



hukaixiang



qwef

qwieufhqef

qwefcqwefcqwecf

wqefc

qwefc

wqefcwefo wefiuh uhwqf

qwefiwqeufh

qwefiuh



qwef

qwefqwef

qwef



Type in your message...



GIF

Send

TCSN • test



Chat



maxmax

You: 😊 7

No groups found :(



hukaixiang



Type in your message...



GIF

Send

qwefi

qwefiwqeufh

qwefiuh

qwef

qwieufhqef



qwef

qwefqwef

qwefcqwefcqwecf

wqefc

qwefc

wqefcwefo wefiuh uhwqf

TCSN • test

Mmaxmax

You: 😊 7

No groups found :(

Hhukaixiang

⚙️

Chat

☀️

qwef

wqefiuh

euhi jdhf ee nbf mnae iuh

wqefi qwuehfiuqwe

qwef

qwieufhqef

qwefi

qwefiwqeufh

qwefiuh

😊

qwef

qwefqwef

😊 |Type in your message...

🖼️

GIF

Send



EXPLORER

REACT-CHAT-REDIS

node\_modules

public

server

config

db.js

mongodb.js

redis.js

controllers

image-upload

models

messages.js

node\_modules

routers

sql

.dockerignore

.env

.gitignore

docker-compose.yml

Dockerfile

index.js

init.sql

package-lock.json

package.json

README.md

src

assets

components

AddFriendModal

TIMELINE

OUTLINE

VS CODE PETS

src > components > Chat > JS index.js > Chat > renderMessageTextWithMention

23 import { getThemeStyles } from "../../components/colorTheme";

24 import { motion } from "framer-motion";

25 import { AccountContext } from "../AccountContext";

26 import ImagePreviewModal from "../ImagePreviewModal";

27 import { Link as ReactRouterLink, useNavigate } from "react-router-

28 // import testImageSrc from "../../assets/images/menu-bg.jpg";

29 import "./styles.scss";

30

31 const Chat = ({ userid, group\_hash, tabIndex }) => {

32   const { friendList } = useContext(FriendContext);

33   const { groups } = useContext(GroupsContext);

34   const { messages } = useContext(MessagesContext);

35   const { currentTheme } = useContext(ThemeContext);

36   const { user } = useContext(AccountContext);

37   const [data, setData] = useState({});

38   const [imagePreviewURL, setImagePreviewURL] = useState("");

39   const theme = getThemeStyles(currentTheme);

40   const bottomDiv = useRef(null);

41   const {

42     isOpen: isImagePreviewModalOpen,

43     onOpen: onImagePreviewModalOpen,

44     onClose: onImagePreviewModalClose,

45   } = useDisclosure();

46   const messageSentBg = useColorModeValue(theme.primaryDark, theme.

47   const messageSentColor = useColorModeValue(theme.lightText, theme

server > JS index.js > ...

100   const filename = Date.now() + path.extname(file.originalname);

101   cb(null, fileName);

102   return fileName;

103   },

104   });

105

106   const upload = multer({

107     storage: storage,

108     // limits: { fileSize: 1024 \* 1024 \* 5 }

109   });

110

111   app.get("/image/:filename", (req, res) => {

112     const filename = req.params.filename;

113     res.sendFile(filename, { root: "../server/image-upload" }, (err) => {

114       if (err) {

115         console.error(err);

116         res.status(404).send("File not found");

117         // res.redirect("https://http.cat/404");

118       }

119     });

120   });

121

122   app.post("/image", limiter, upload.single("image"), (req, res) => {

123     if (!req.file) {

124       return res.status(400).send("No image uploaded");

125     }

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

COMMENTS

node - server

node

node server

connected: 'true'

hukaixiang connected [

{

username: 'maxmax',

userid: '02117cc3-354a-412b-9f6f-8db11f13e0c0',

connected: 'true'

}

]

A user disconnected

A user disconnected

webpack compiled successfully

Note that the development build is not optimized.

To create a production build, use npm run build.

You can now view tcsn in the browser.

Local: http://localhost:3000

On Your Network: http://192.168.1.104:3000

Compiled successfully!



README.md M

init.sql X



server > init.sql

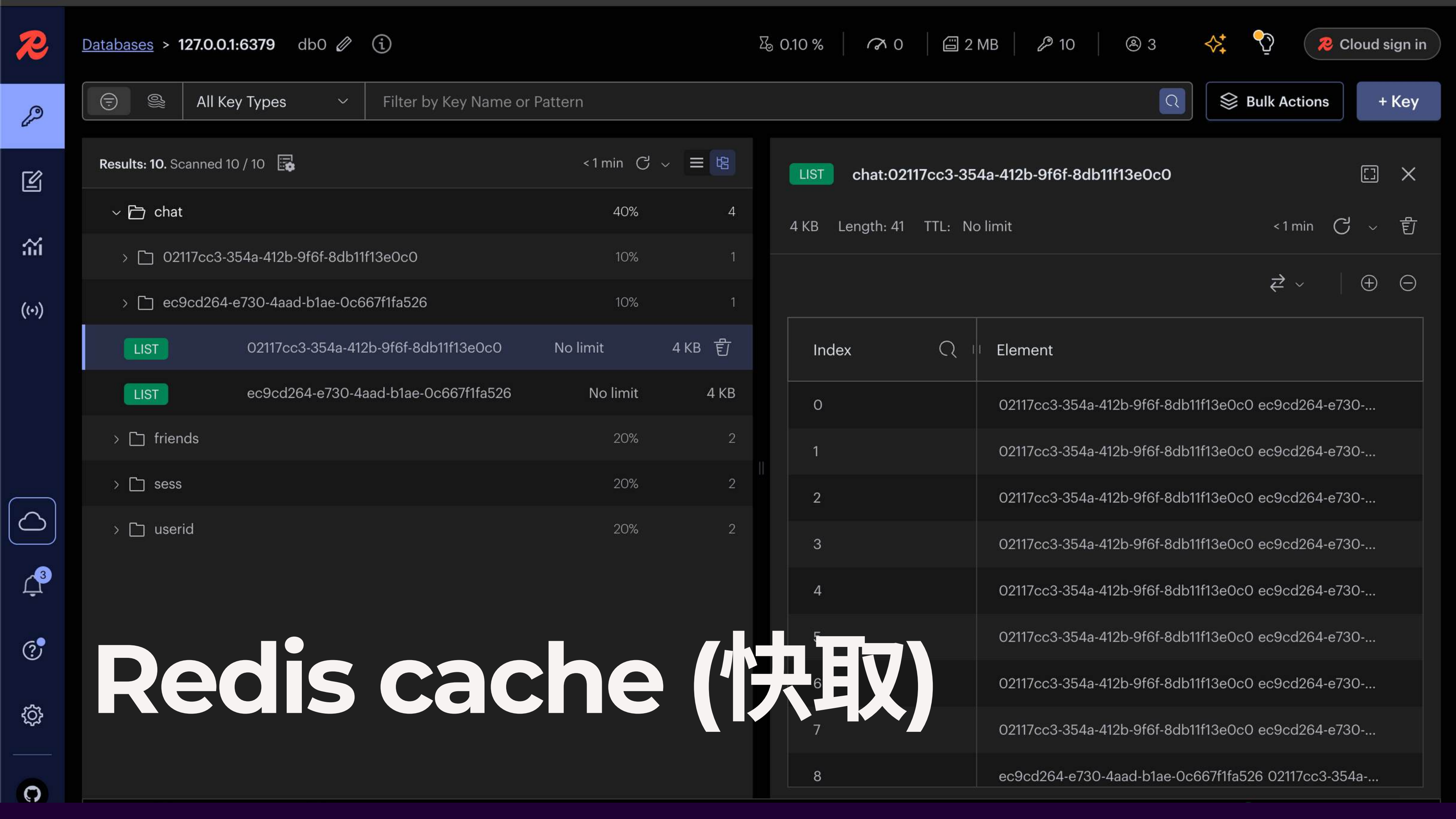
You, 5 months ago | 1 author (You)

```
1
2 CREATE TABLE users(
3     id SERIAL PRIMARY KEY,
4     username VARCHAR(32) NOT NULL UNIQUE,
5     passhash VARCHAR NOT NULL,
6     userid VARCHAR NOT NULL UNIQUE,
7     bio VARCHAR,
8     profile_picture VARCHAR
9 );
```

You, 5 months ago • working docker

# DB Schema

資料庫綱要



Redis cache (快取)

DEMO



**THANKS FOR  
LISTENING**