Authentication / Authorization

Part 4: SPA

Setup

- git clone -b main https://github.com/fullstack-67/auth-spa.git auth-spa
- Backend
 - cd backend
 - Fill in .env
 - o pnpm install
 - npm run db:reset
 - o npm run dev

Setup

- Frontend
 - o cd frontend
 - o pnpm install
 - o npm run dev

Backend

- Routes now return json / redirect header instead of html.
- Created GET /me route for clients to check their auth states.
- Apart from that, there is very mininal change, surprisingly.

```
∠ auth-spa

                                                                                                                                                                        File Edit Selection View Go Run Terminal Help
                                                                                                                                                                           43 ↑ ↓ Z ¶ Ø □ ···
               index.mpa.ts ↔ index.ts X
                backend > src > 15 index.ts > ...
                22 hidden lines
                                                                                                           22 hidden lines
                 23 app.use(passportIns.session());
                                                                                                           23 app.use(passportIns.session());
                                                                                                           25 // * Endpoints
                                                                                                           26+ app.get("/me", async (req, res, next) => {
                 26-app.get("/", async (req, res, next) => {
                 27    const sessions = await formatSession(req);
                                                                                                                const sessions = await formatSession(req);
                                                                                                                const user = req?.user ?? null;
                 res.render("pages/index", {
                        title: "Home",
         user: req.user,
                                                                                                                res.json({ sessions, user });
                        sessions: sessions,
                 35 app.get("/signup", function (req, res) {
                                                                                                           32 app.get("/signup", function (req, res) {
                34 hidden lines
                                                                                                           70 app.post("/login", passportIns.authenticate("local"), function (req, res) {
                                                                                                           67 app.post("/login", passportIns.authenticate("local"), function (req, res) {
                      debug("@login handler");
                                                                                                                 debug("@login handler");
                      setSessionInfoAfterLogin(req, "CREDENTIAL");
                                                                                                                 setSessionInfoAfterLogin(req, "CREDENTIAL");
                      res.setHeader("HX-Redirect", "/");
                                                                                                                res.status(200).json("Login Successful");
                 74- res.send(`<div></div>`);
                 app.get("/login/oauth/github", passportIns.authenticate("github"));
                                                                                                           73 app.get("/login/oauth/github", passportIns.authenticate("github"));
                20 hidden lines | 🕥 app.get("/callback/google") callback
                                                                                                           20 hidden lines | 🕥 app.get("/callback/google") callback
                                                                                                           95
                101— app.post("/logout", function (req, res, next) {
                                                                                                           97+ app.get("/logout", function (req, res, next) {
                      // req.logout will not delete the session in db. It will generate new one for the
                                                                                                                // req.logout will not delete the session in db. It will generate new one for the alro
                                                                                                                 // When the user login again, it will generate new session with the user id.
                      // When the user login again, it will generate new session with the user id.
                      req.logout(function (err) {
                                                                                                                 req.logout(function (err) {
                $ 5 hidden lines | ♦ req.session.destroy() callback
                                                                                                          $ 5 hidden lines | 😚 req.session.destroy() callback
                          if (err) {
                                                                                                                     if (err) {
                            return next(err);
                                                                                                                      return next(err);
                          res.setHeader("HX-Redirect", "/");
                                                                                                                    res.redirect("/");
                          res.send("<div></div>");

★ 13 hidden lines

★ 13 hidden lines

261497 Füllstäck Development
```

Frontend

- Created (client-side) routing.
- Created logic to query/update auth 's state.
- Modified proxy server to take care of Callback URL (bypass client-routing).
- Created signup form/login and other Ul.

Highlighted packages

```
{
   "@tanstack/react-query": "^5.52.1",
   "react-router-dom": "^6.26.1"
}
```

9

Client-side routing

src/App.tsx

```
import { createBrowserRouter, RouterProvider } from "react-router-dom";
const router = createBrowserRouter([
    path: "/",
    element: <Layout />,
    children: [
        path: "/",
        element: <Home />,
      },
```

Client-side routing

```
src/App.tsx
```

Getting auth state

src/hooks/useAuth.ts

```
import { useQuery } from "@tanstack/react-query";
// ...
function getMe() {
  return axios.get<AuthData>("/api/me");
function useAuth() {
  // Queries
  const { data, error, refetch } = useQuery({
    queryFn: getMe,
    // Other options
  });
  return { user: data?.user, sessions: data?.sessions, error, refetch };
```

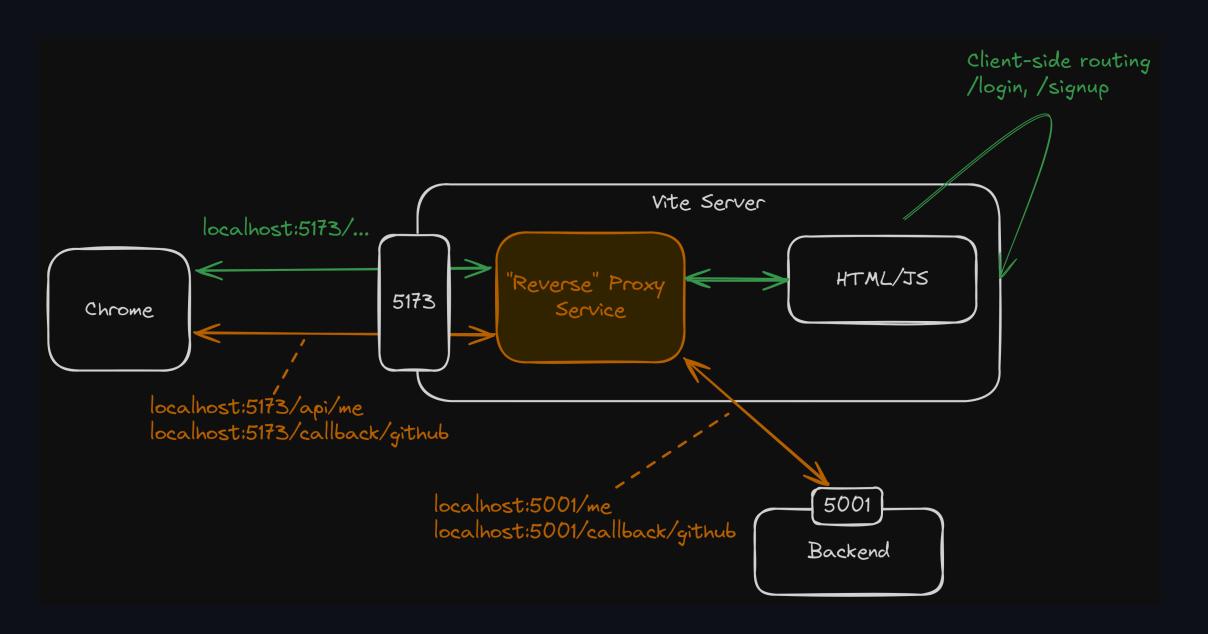
Getting auth state

src/components/Nav.tsx

Handle Callback URL

vite.config.ts

```
export default defineConfig({
  plugins: [react()],
  server: {
    proxy: {
     "/api": {
      // ...
      "/callback": { 🍑 🖜
       target: "http://localhost:5002",
      },
```



If you don't have proxy server.

- Cookies are still automatically sent given the same host but different port.
- Need to set withCredentials to true on any AJAX request (via fetch or axios APIs)
 - Note that http0nly cookie are sent by this technique.
- Need to allow CORS in the backend.

16

Auth.js

Setup

- git clone -b authjs https://github.com/fullstack-67/auth-mpa-v2.git auth-authjs
- pnpm i
- npm run db:reset
- npm run dev

Note

- DB schema
- Available auth methods
- Check cookie after login.