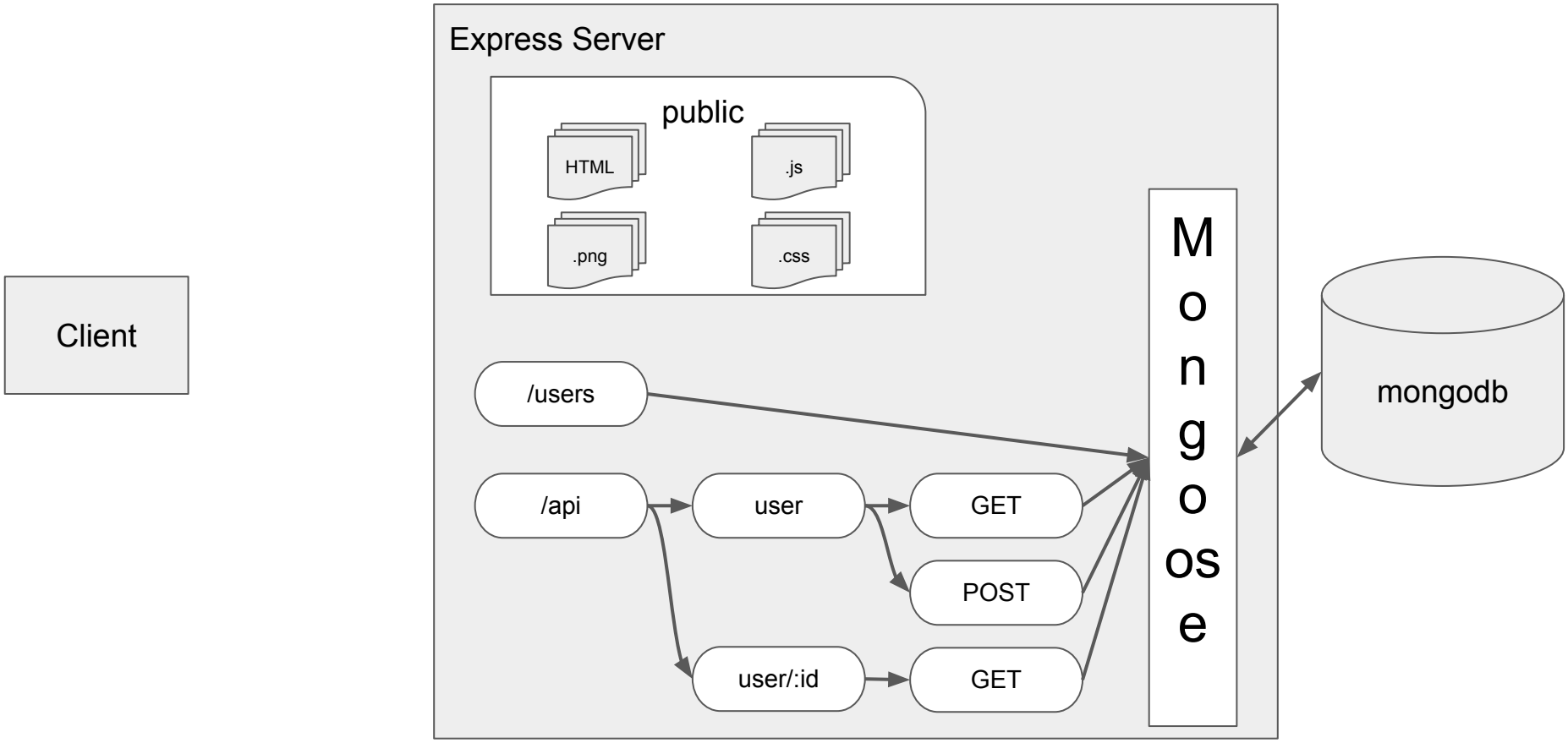
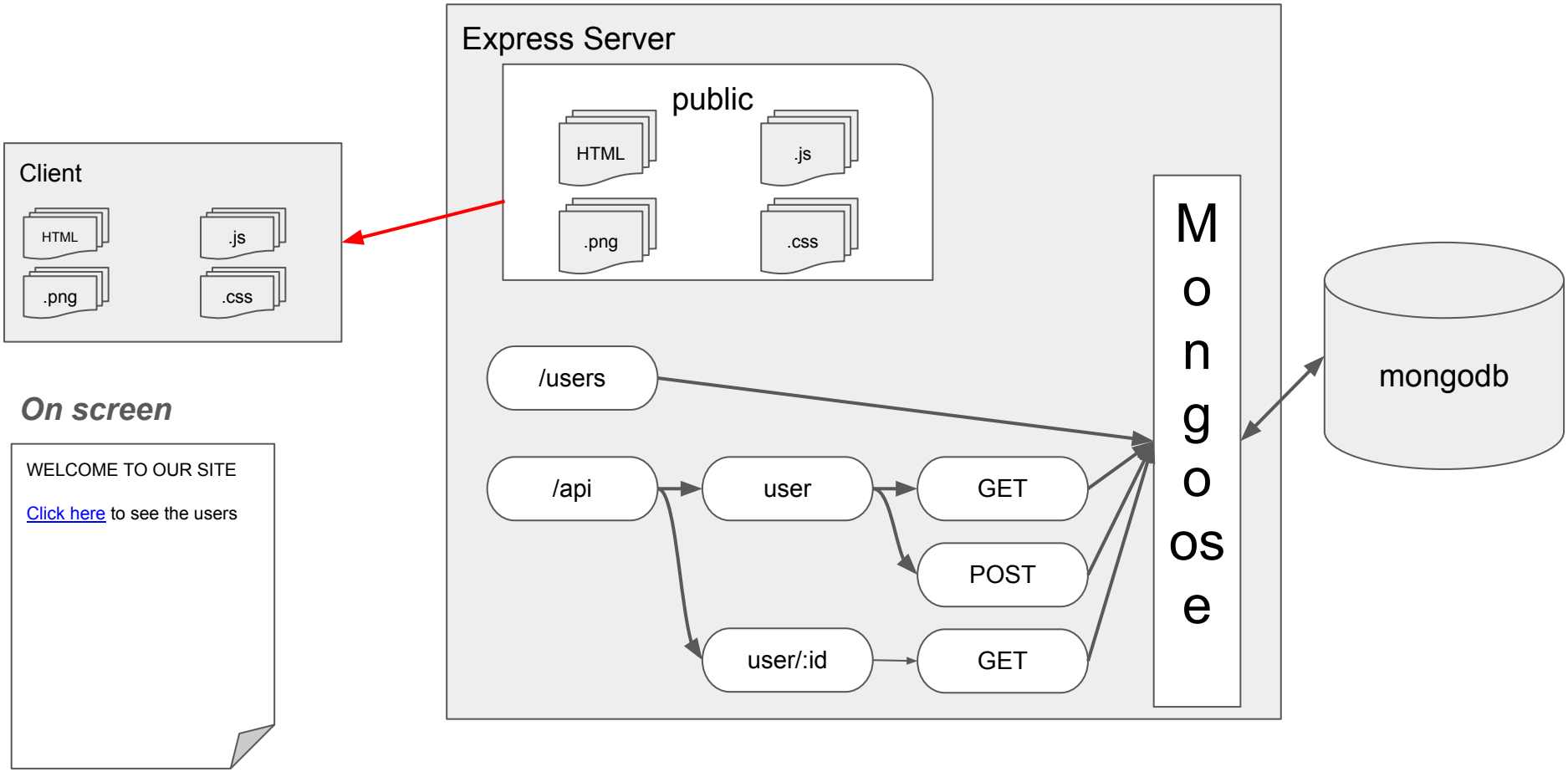
Server Side Rendered

E.g., Handlebars


Client
Express Server
public
HTML
.js
.png
.css
/users
/api
user
GET
POST
user/:id
GET
Mongoose
mongodb

2. Client Requests `http://localhost:3000/`

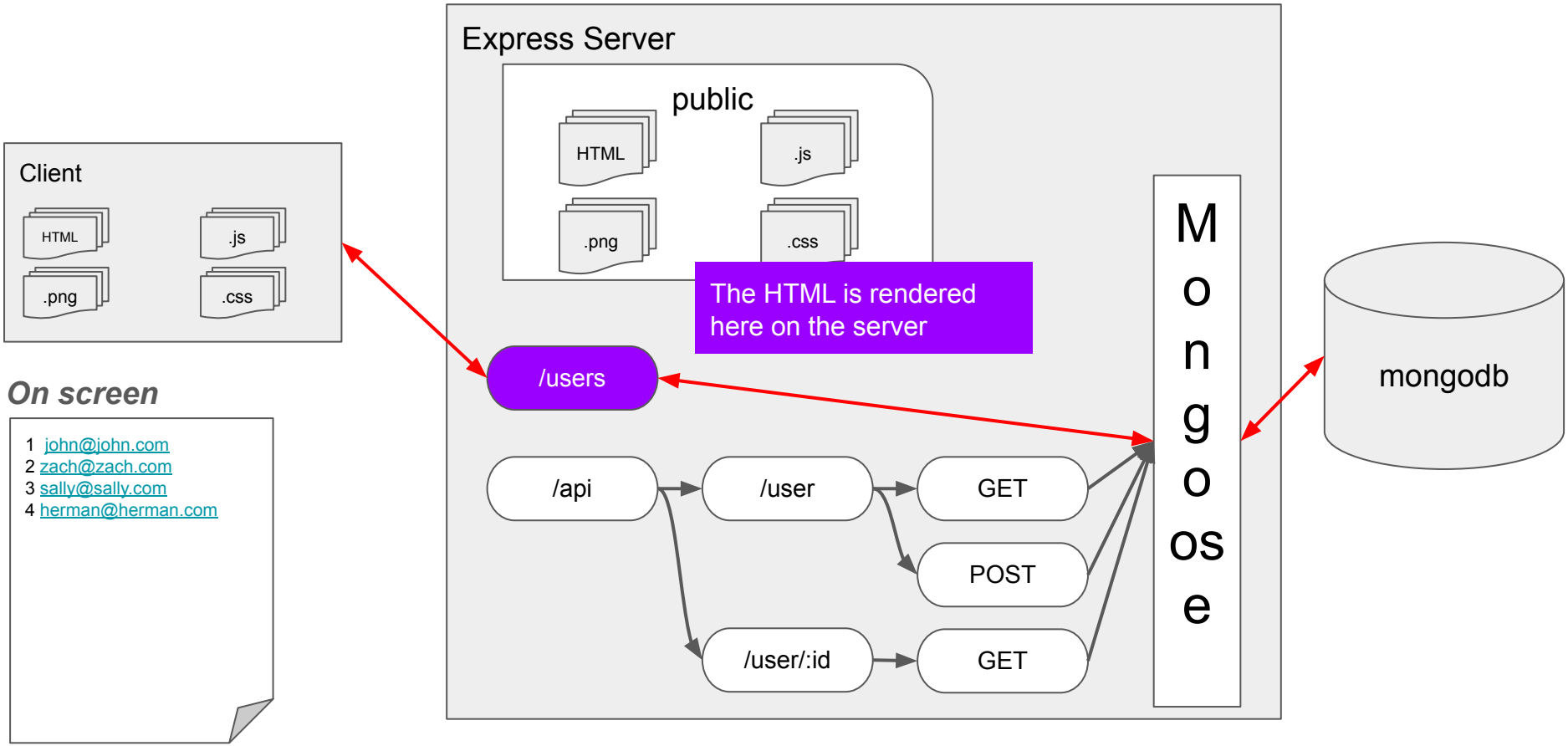


On screen

WELCOME TO OUR SITE

Click here to see the users

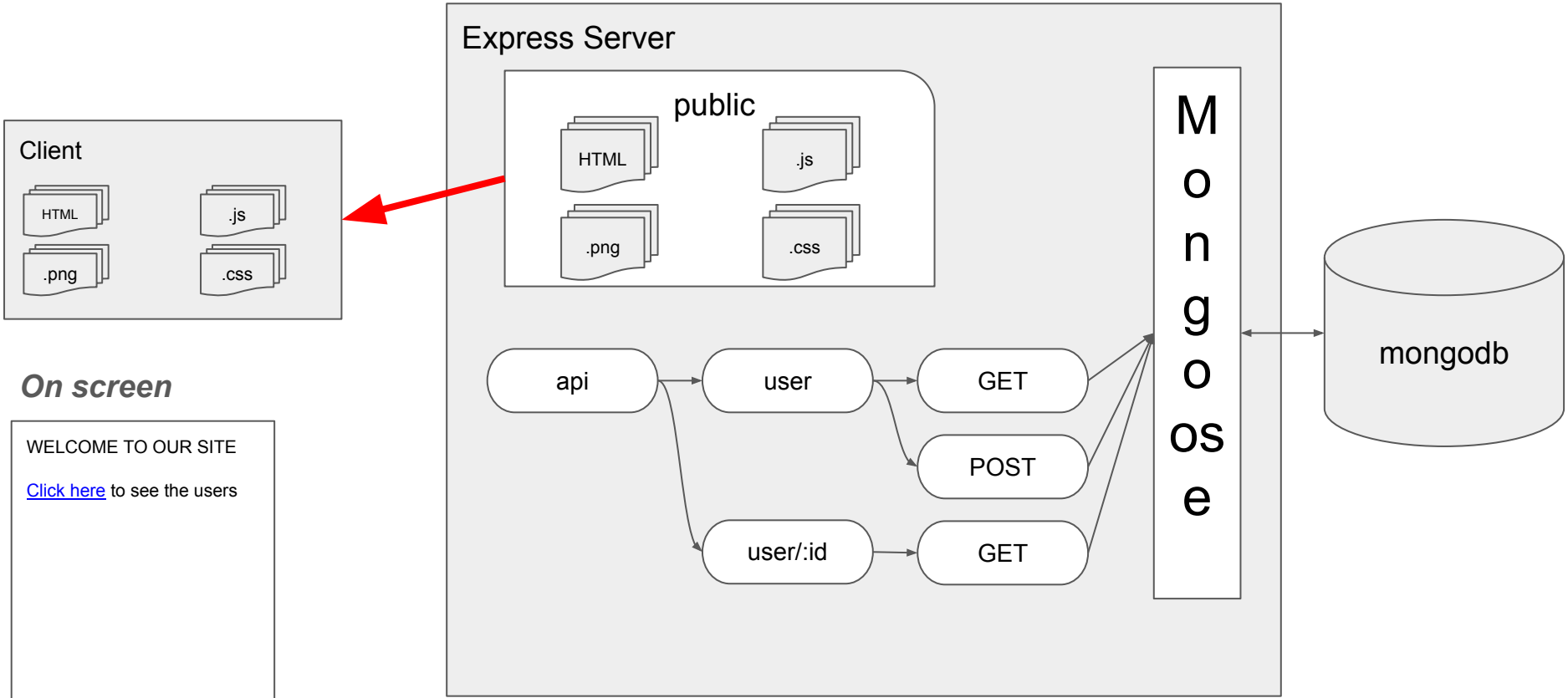
2. Client Requests `http://localhost:3000/users`



Single Page App

Client side rendered HTML using DOM manipulation

1. Client Loads Front End

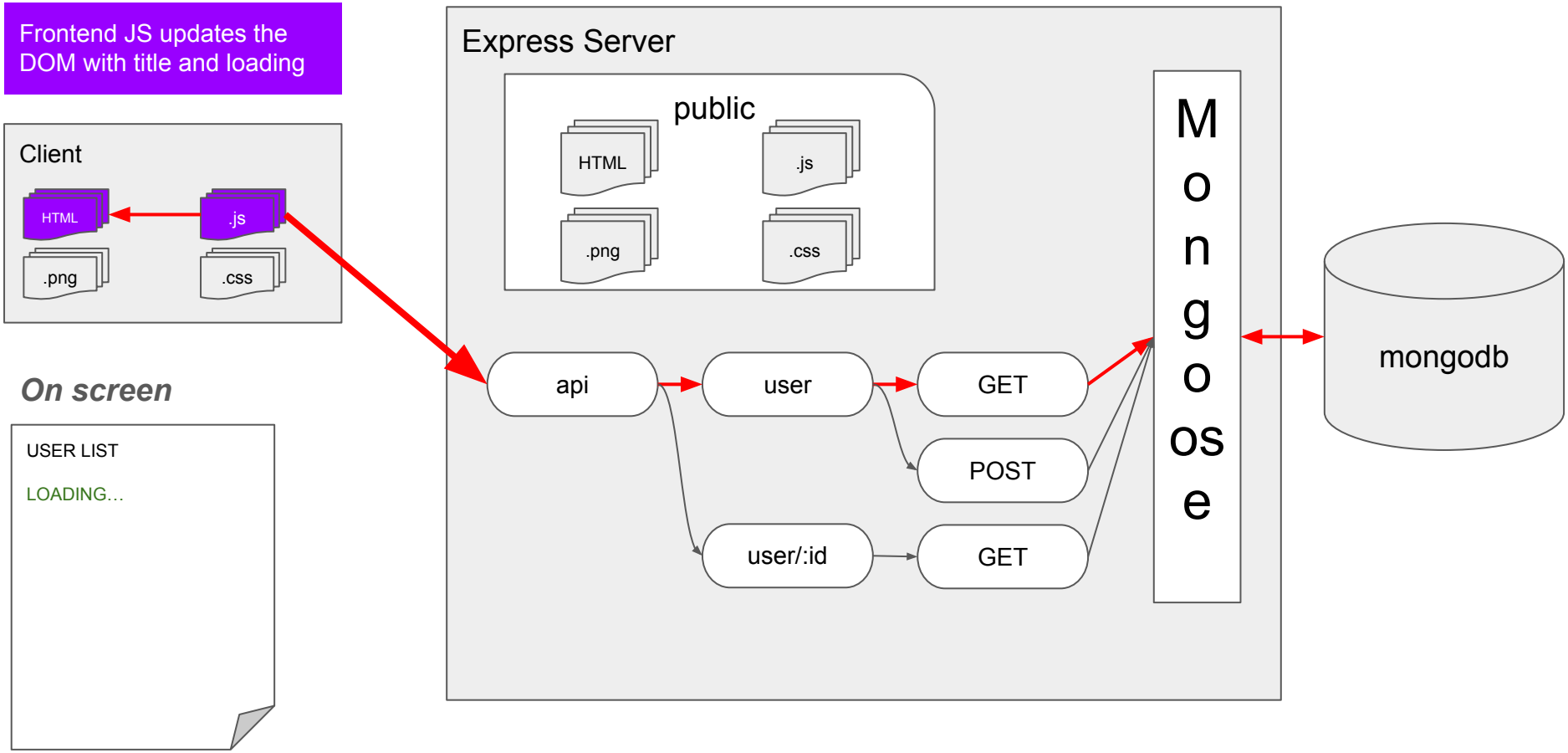


On screen

WELCOME TO OUR SITE

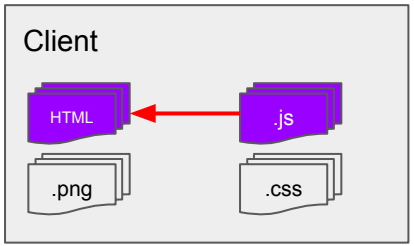
Click here to see the users

2. Frontend JS requests list of users (still at http://localhost:3001)



3. Frontend JS updates the DOM, screen shows user list

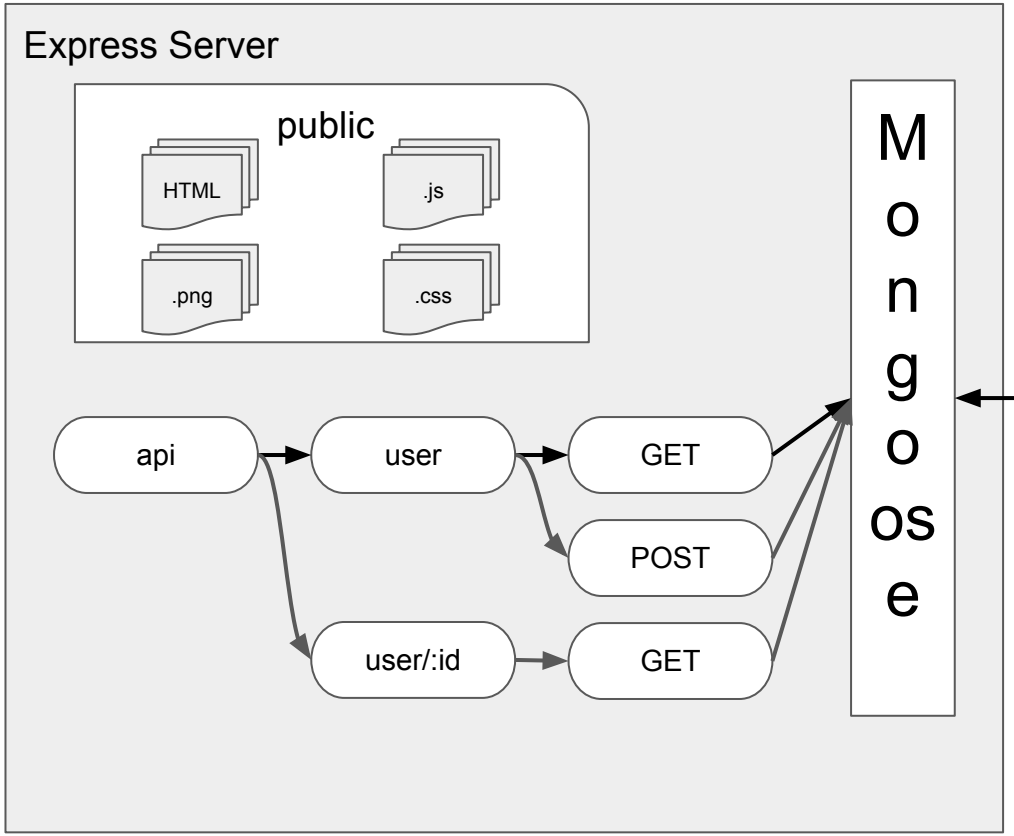
User list HTML is rendered here on the client

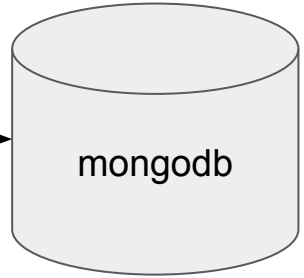


On screen

USER LIST

1 john@john.com
2 zach@zach.com
3 sally@sally.com
4 herman@herman.com

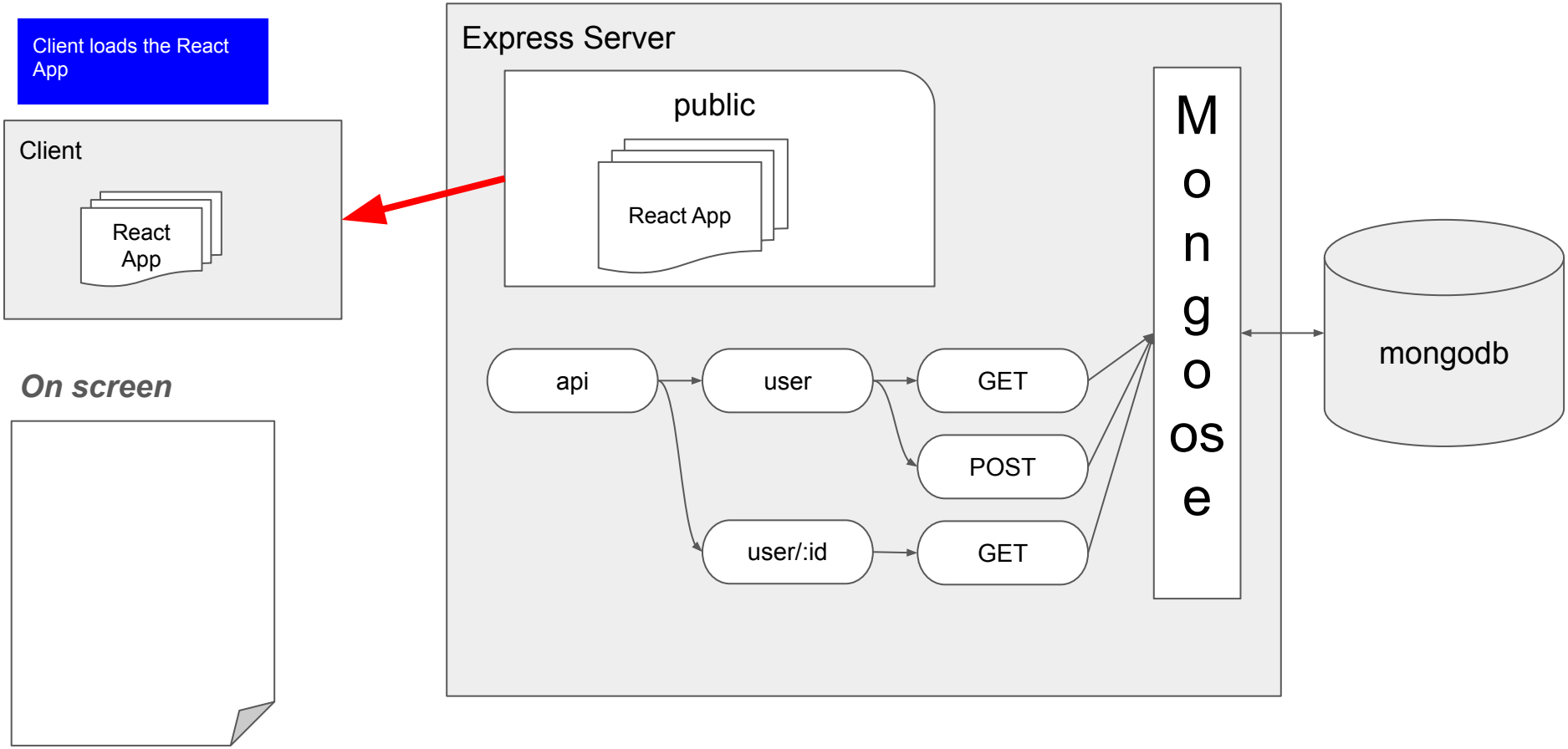




Single Page App

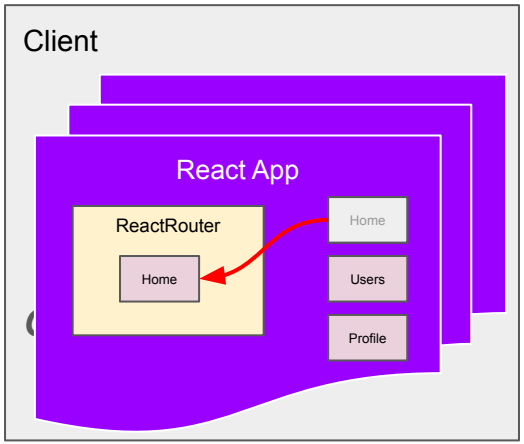
Client side rendered React

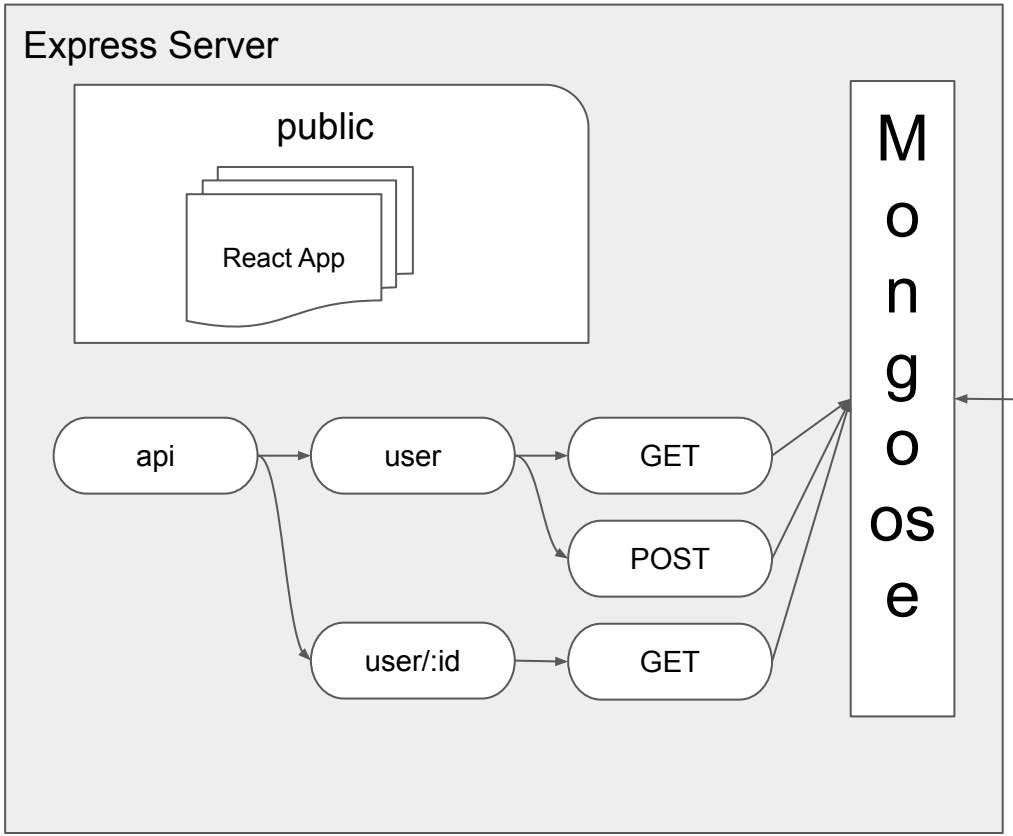
1. Client Requests `http://localhost:3000/`



2. React renders Home page

ReactRouter renders the Home component because the path / points to Home



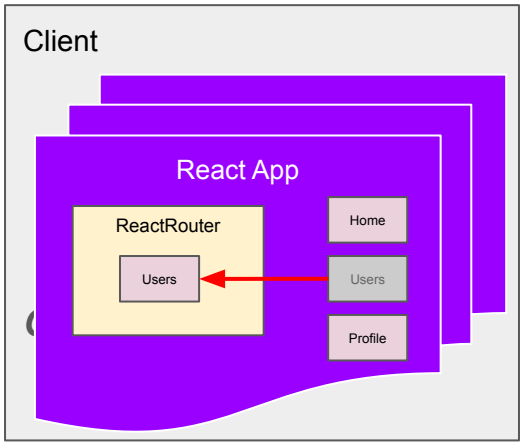


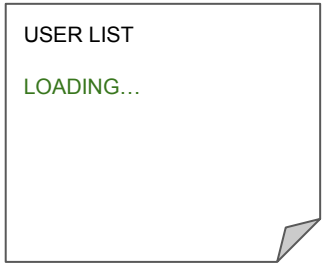
WELCOME TO OUR SITE

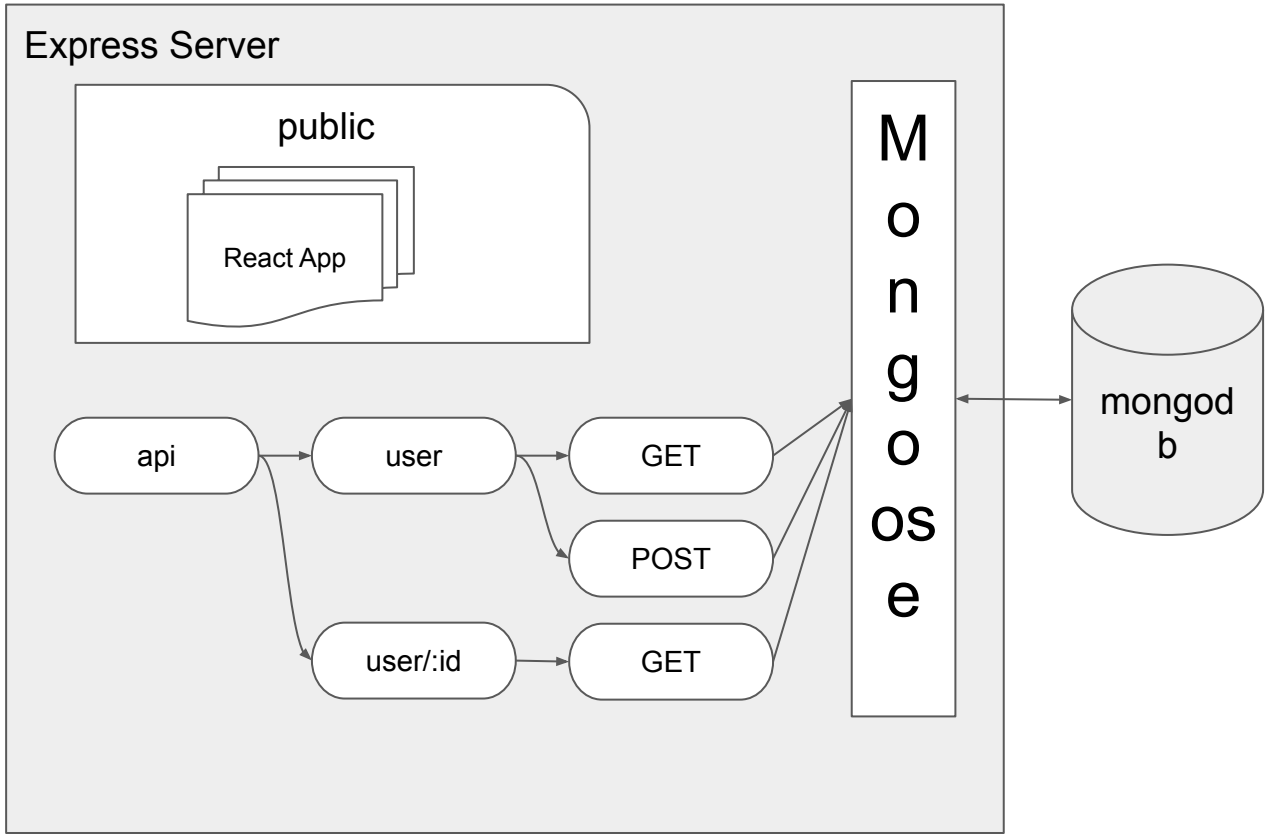
Click here to see the users

2. Client Requests `http://localhost:3000/users`

ReactRouter renders the Users component because the path /users points to User

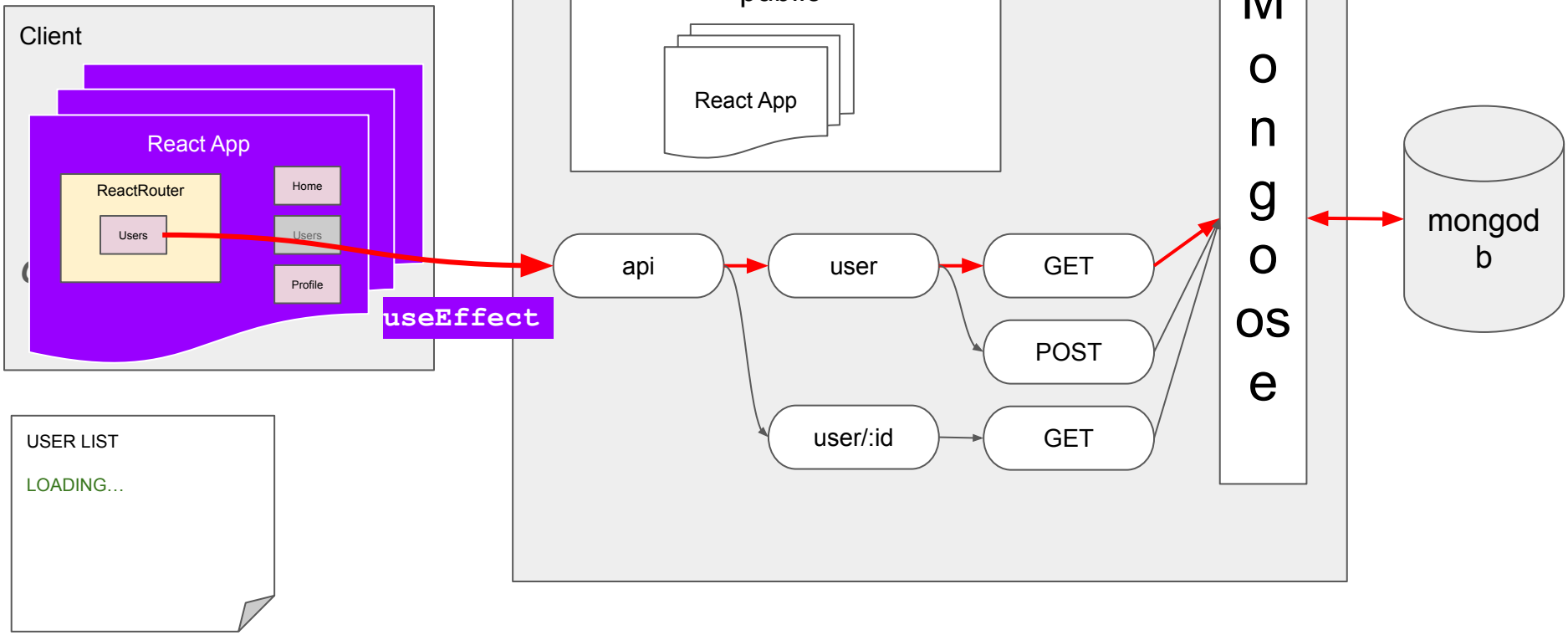






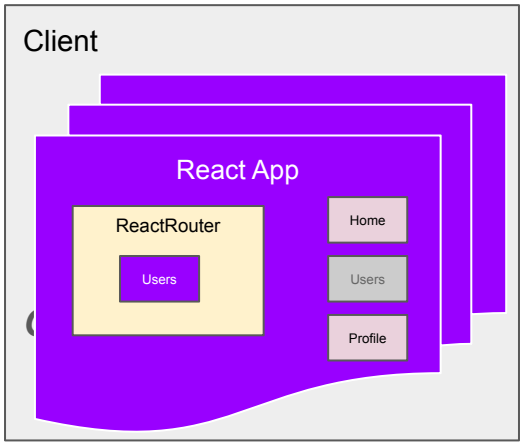
3. Users component useEffect requests the list of users and update state

ReactRouter renders the Users component because the path /users points to User



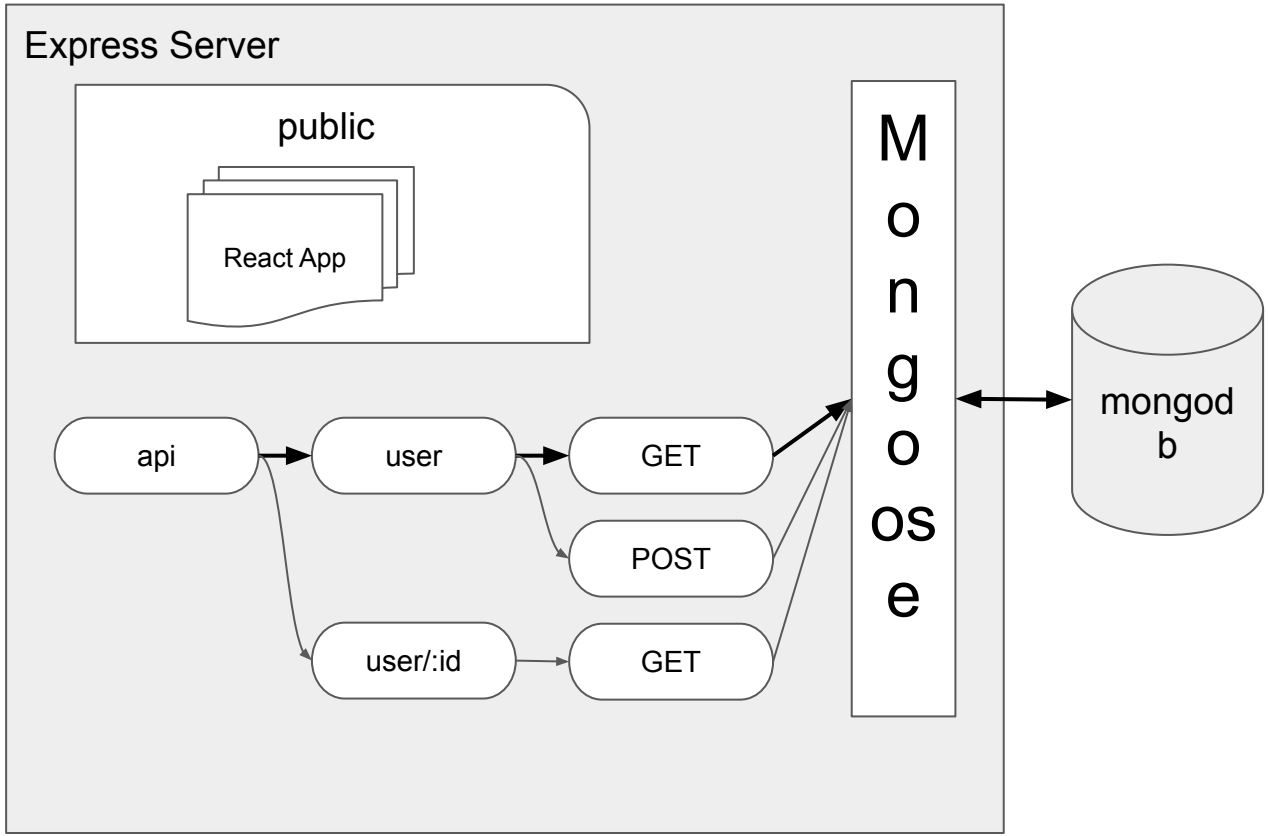
3. Users component re-renders because state has changed

ReactRouter renders the Users component because the path /users points to User



USER LIST

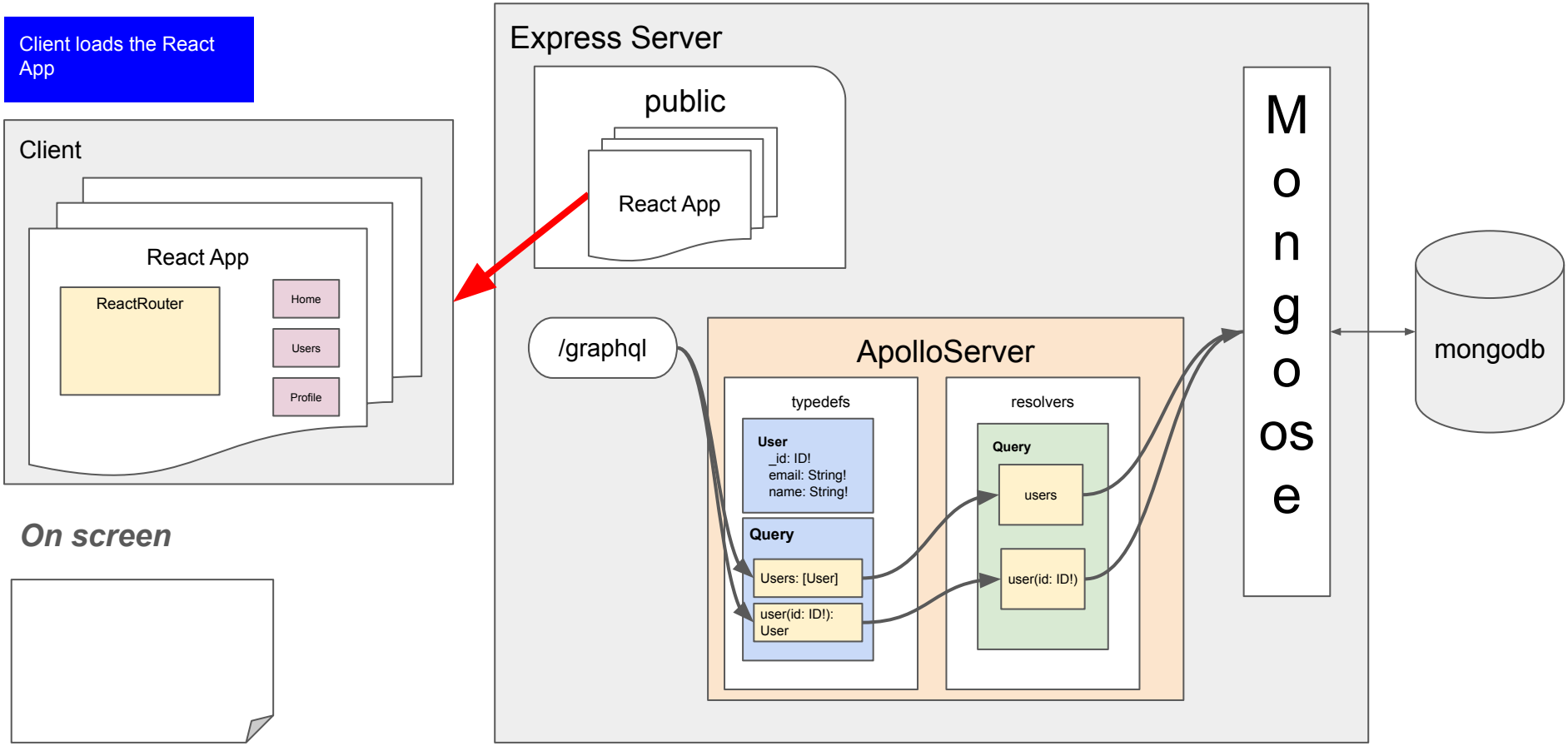
1 john@john.com
2 zach@zach.com
3 sally@sally.com
4 herman@herman.com



Single Page App

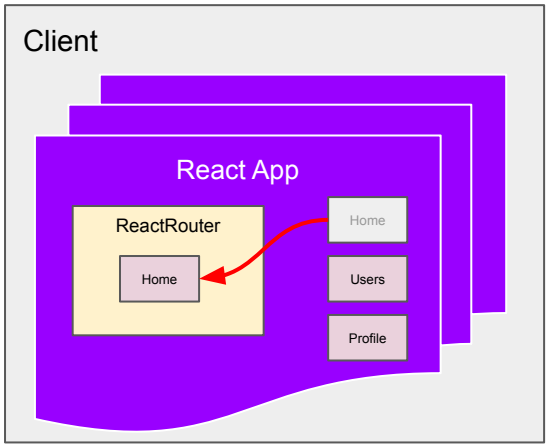
GraphQL

1. Client Requests `http://localhost:3000/`



2. React renders Home page

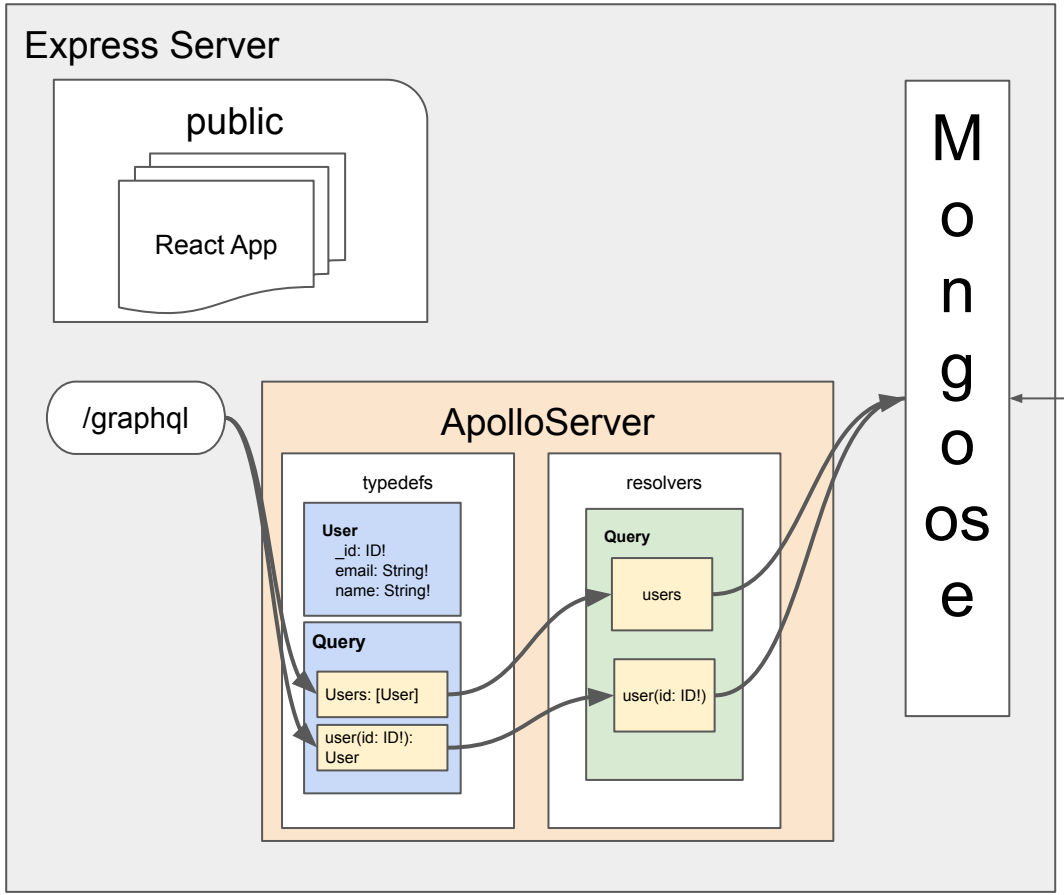
ReactRouter renders the Home component because the path / points to Home



On screen

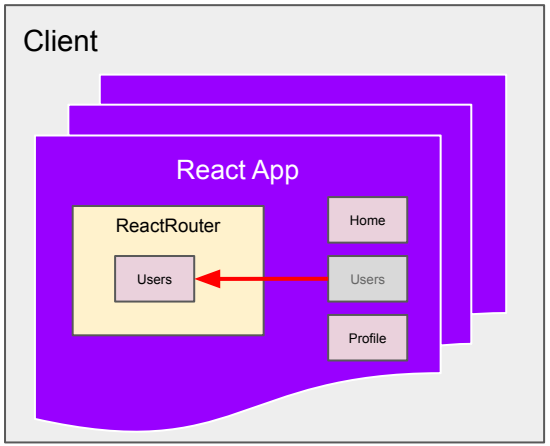
WELCOME TO OUR SITE

Click here to see the users



3. Client Requests `http://localhost:3000/users`

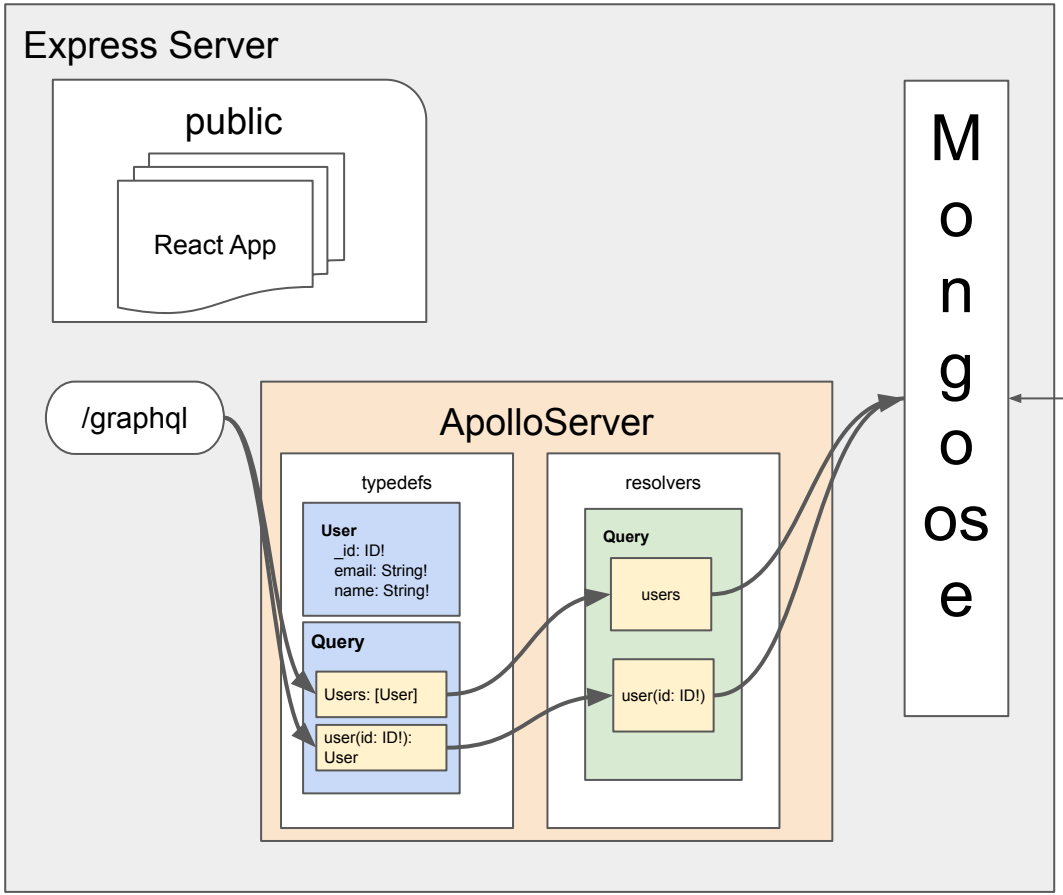
ReactRouter renders the User component because the path /users points to User -



On screen

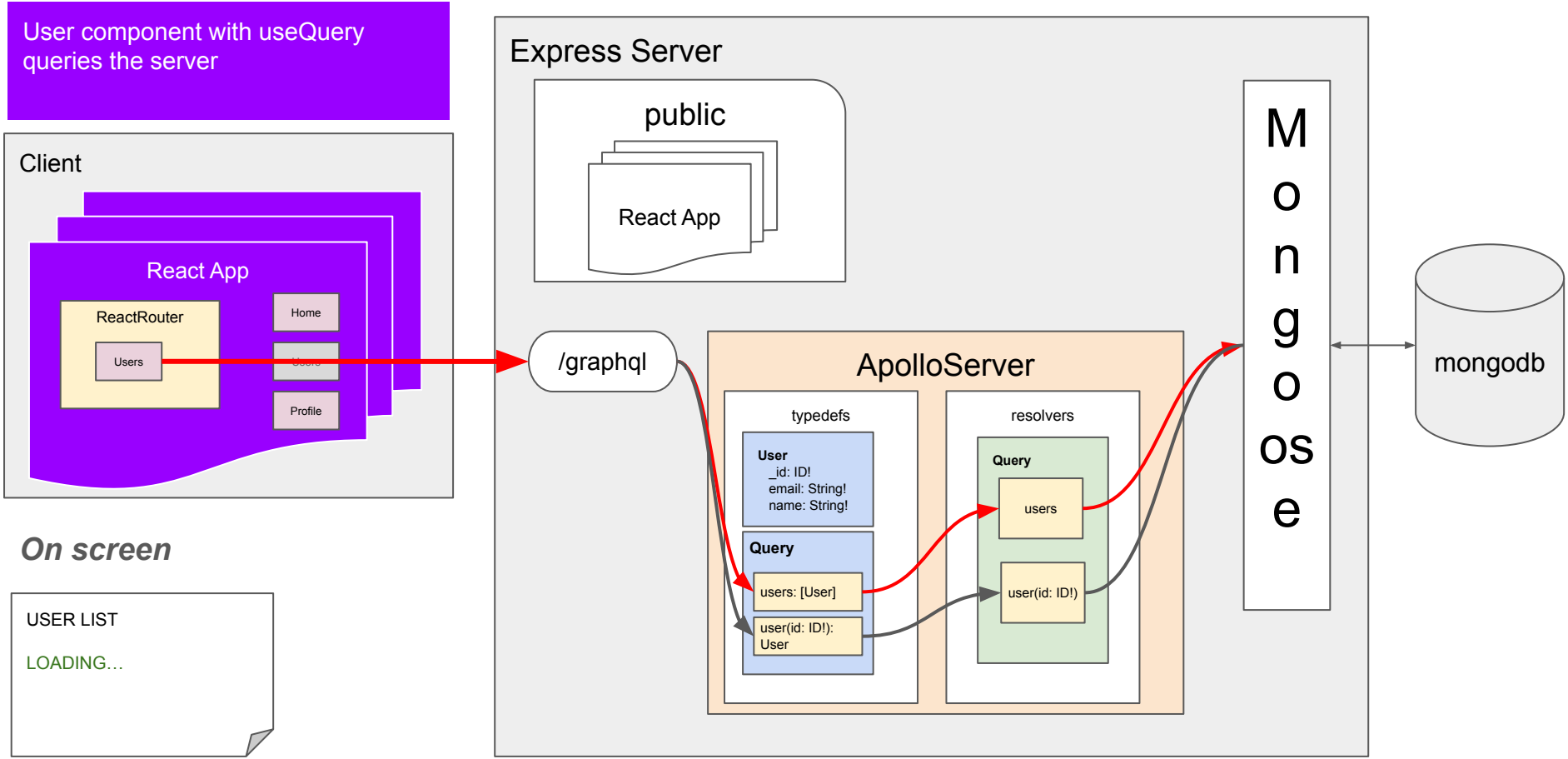
USER LIST

LOADING…



4. Users component useQuery requests the list of users effectively updating state

User component with useQuery queries the server



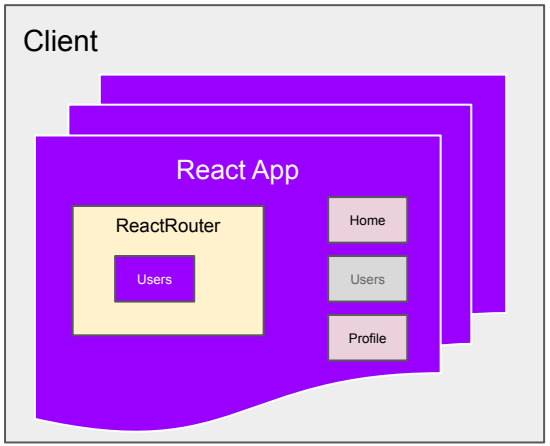
On screen

USER LIST

LOADING…

5. User component re-renders

User component re-renders



On screen

USER LIST

1 john@john.com
2 zach@zach.com
3 sally@sally.com
4 herman@herman.com