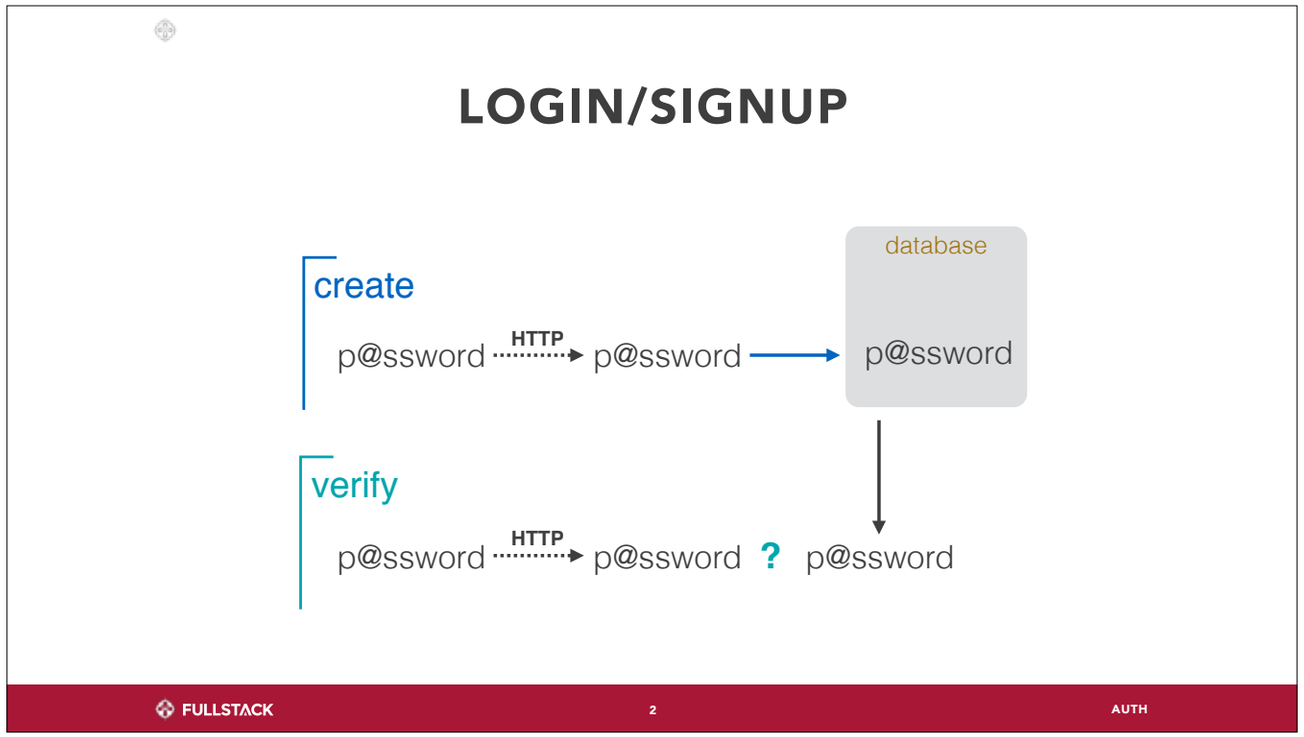
LOGIN AND SIGNUP

That's amazing! I've got the same combination on my luggage!

Let's talk about authentication data flow. First of all, disclaimer: we're not talking about web security today. We're just going to talk about the flow of data involved in logging in users and keeping them logged in.

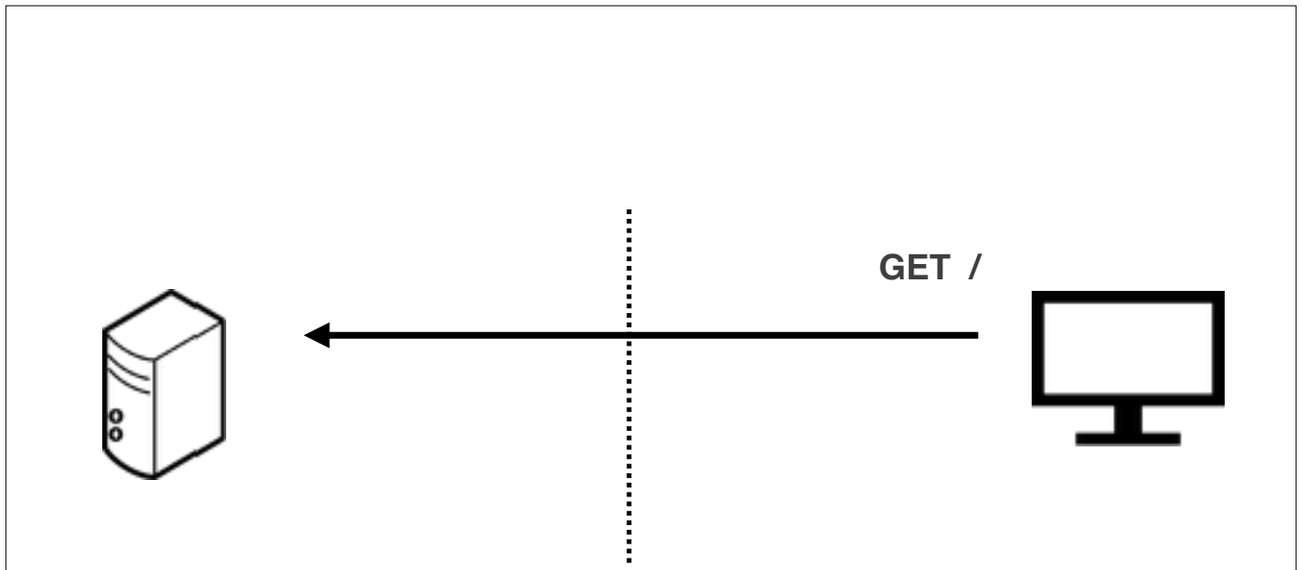


Here’s a general outline of the login/signup workflow, at least the way we understand it right now. It’s more or less what you’d expect.

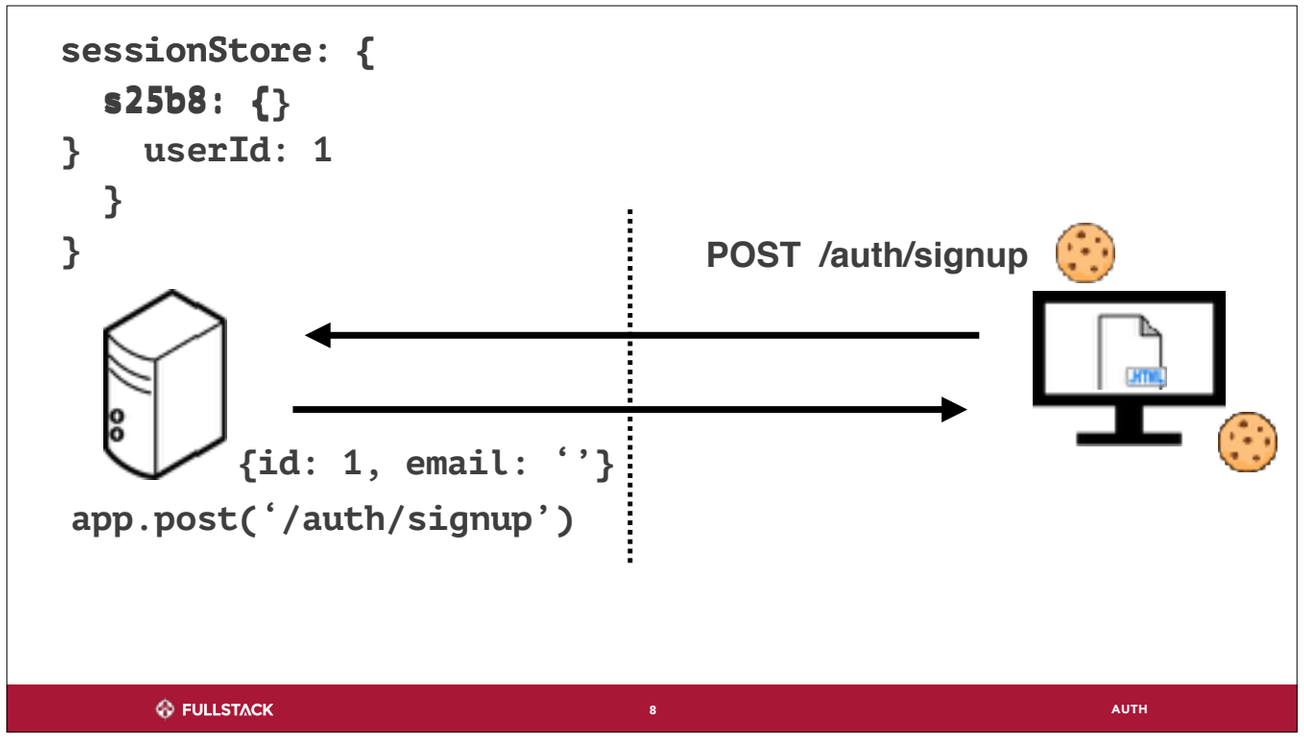
OPERATIONS

- GET /me (*who am I?*)
- POST /signup (*let me introduce myself*)
- PUT /login *(you remember me, right?)*
- DELETE /logout (*please just forget me*)

SIGNUP/LOGIN


GET /

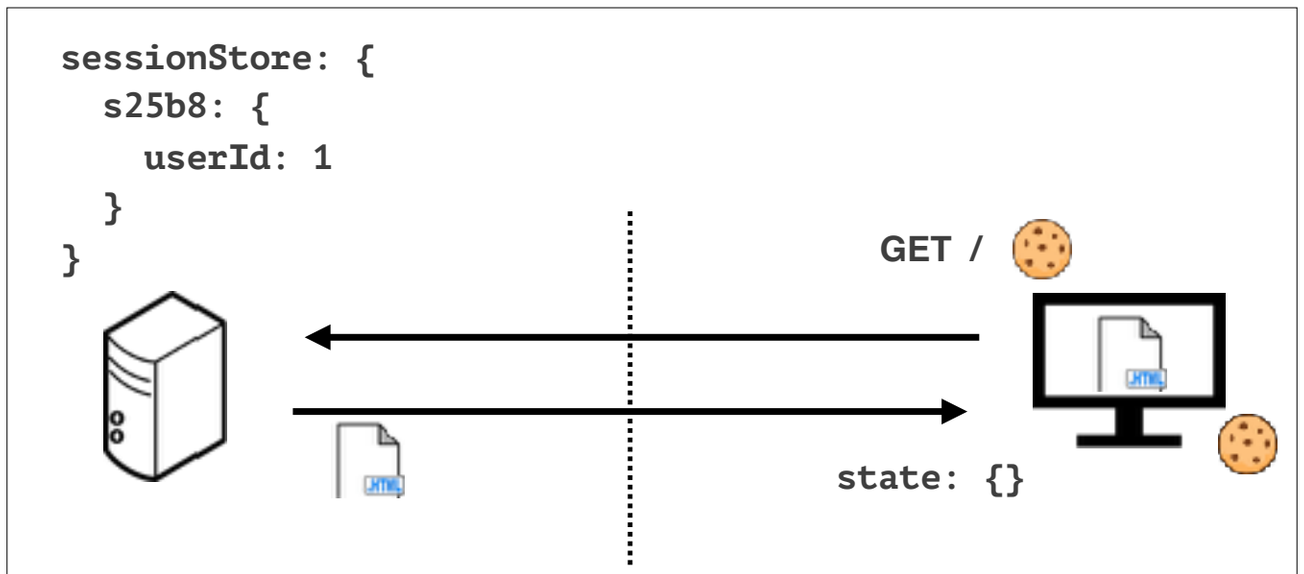
```
sessionStore: {
  s25b8: {}
}
```


sessionStore: {
s25b8: {}
} userId: 1
}
}
POST /auth/signup
{id: 1, email: ‘’}
app.post(‘/auth/signup’)

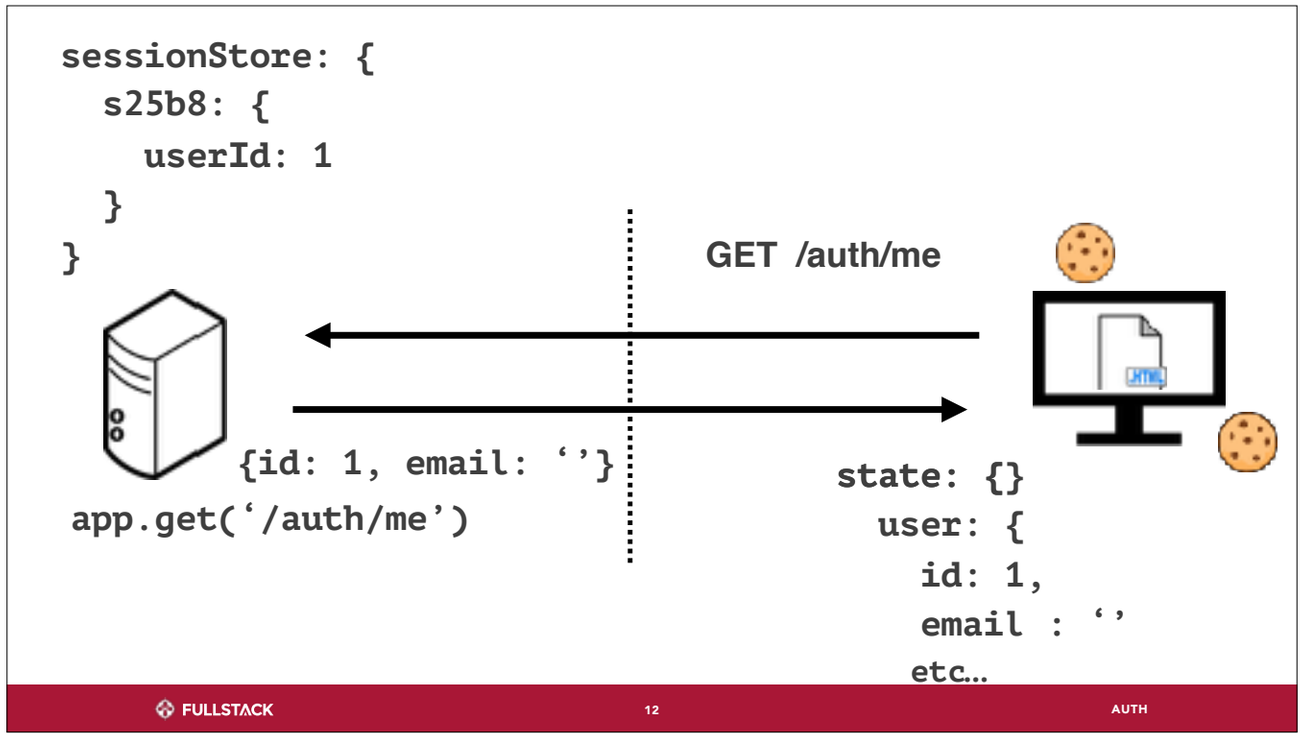
PERSISTING AFTER REFRESH

```
sessionStore: {
  s25b8: {
    userId: 1
  }
}
```

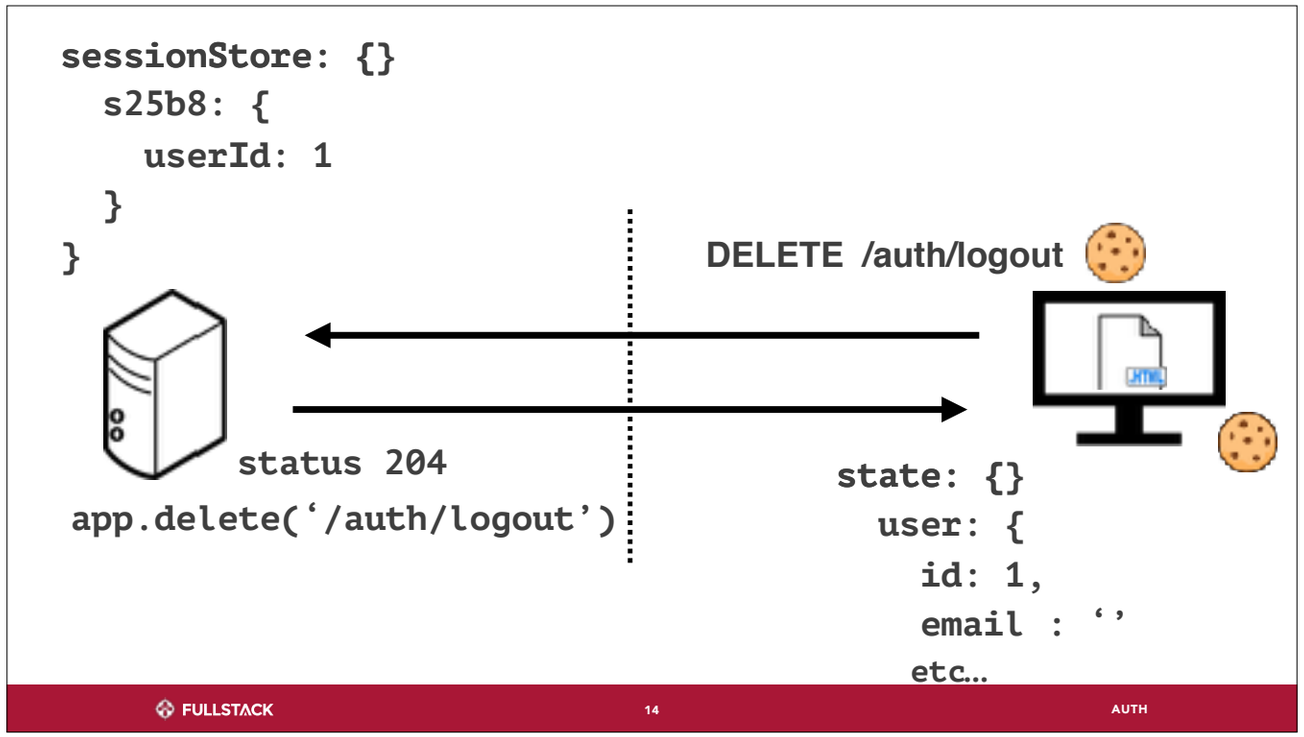

```
state: {
  user: {
    id: 1,
    email : ‘’
    etc…
```



First we need to get our index html back, before we can do anything


sessionStore: {
s25b8: {
userId: 1
}
}
GET /auth/me
{id: 1, email: ‘’}
app.get(‘/auth/me’)
state: {}
user: {
id: 1,
email : ‘’
etc...

LOGOUT


sessionStore: {}
s25b8: {
userId: 1
}
}
DELETE /auth/logout
status 204
app.delete('/auth/logout')
state: {}
user: {
id: 1,
email : ''
etc...