Authentication Authorization and Security

week7

Security

- Data security is important in online business
- Backend is the part of a website which is hidden from the user: DB, application, server
- When we deal with sensitive data (user data), it needs to be carefully dealt
- Authentication deals with who s/he is (verifying a user) (Ex: login/password)
- Authorization checks the status of him/her (granting access and privileges) (Ex: admin, manager, superuser, etc.)

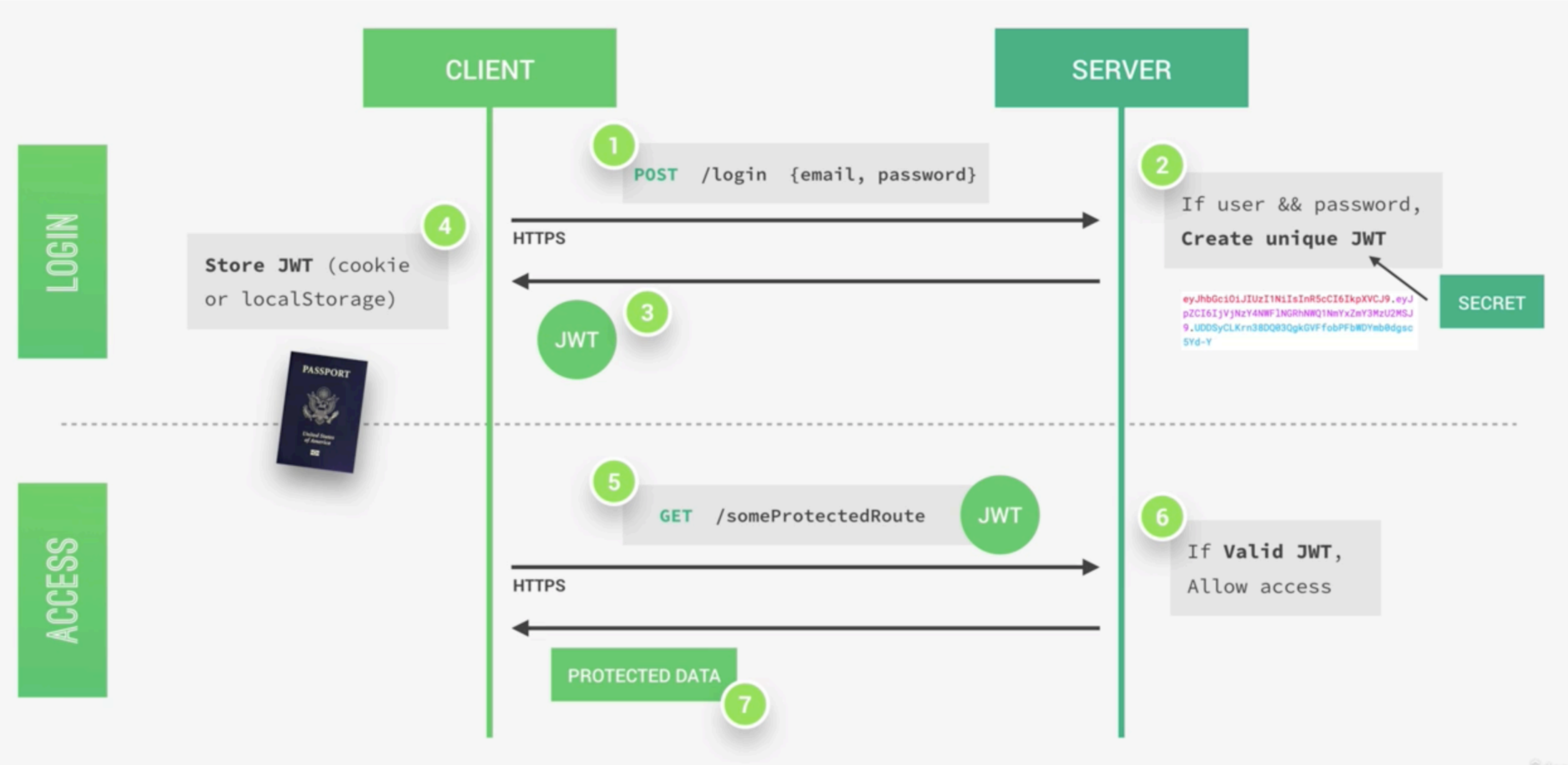
Authentication vs Authorization

Authentication	Authorization
Determines whether users are who they claim to be	Determines what users can and cannot access
Challenges the user to validate credentials (for example, through passwords, answers to security questions, or facial recognition)	Verifies whether access is allowed through policies and rules
Usually done before authorization	Usually done after successful authentication
Generally, transmits info through an ID Token	Generally, transmits info through an Access Token
Example: Employees in a company are required to authenticate through the network before accessing their company email	Example: After an employee successfully authenticates, the system determines what information the employees are allowed to access

Modelling Users

- in the models folder create userModel.js
- create a userModel schema and fill up with important user info
- create authController.js in controllers folder
- create catchAsync.js in utils folder to automate the async calls
- password check and encryption (npm i bcryptjs)

Authentication with JsonWebTokens (JWT)



CLIENT

SERVER

LOGIN

1 POST /login {email, password}

HTTPS

2 If user && password,
Create unique JWT

SECRET

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
pZCI6IjVjNzY4NWFlNGRhNWQ1NmYxZmY3MzU2MSJ
9.UDDSyCLKrn38DQ03QgkGVFfobPFbWDYmb0dgsc
5Yd-Y
```

3 JWT

4 Store JWT (cookie or localStorage)

ACCESS

5 GET /someProtectedRoute JWT

HTTPS

6 If Valid JWT,
Allow access

PROTECTED DATA 7

Authentication

- npm i jsonwebtoken
- login users
-