HackYourFutureBE
Digital Empowerment Project

Week 5
Authentication & Authorization

December 2, 2018

Retrospective

This week

Logging users in/out

Securing routes server-side

Securing user actions client-side

Why both?

Authentication vs. Authorization

Deciding whether someone is who they say the are (and logging them in, for example)

Deciding who is allowed to do what. For example, an admin can send a newsletter, while a regular user can’t.

JSON web token

- Securely transmit information between computers as a JSON object
- NOT encrypted by default, but signed to ensure integrity of a claim

We aren’t concerned with hiding the claims users make,
only that they have the authority to make that claim.

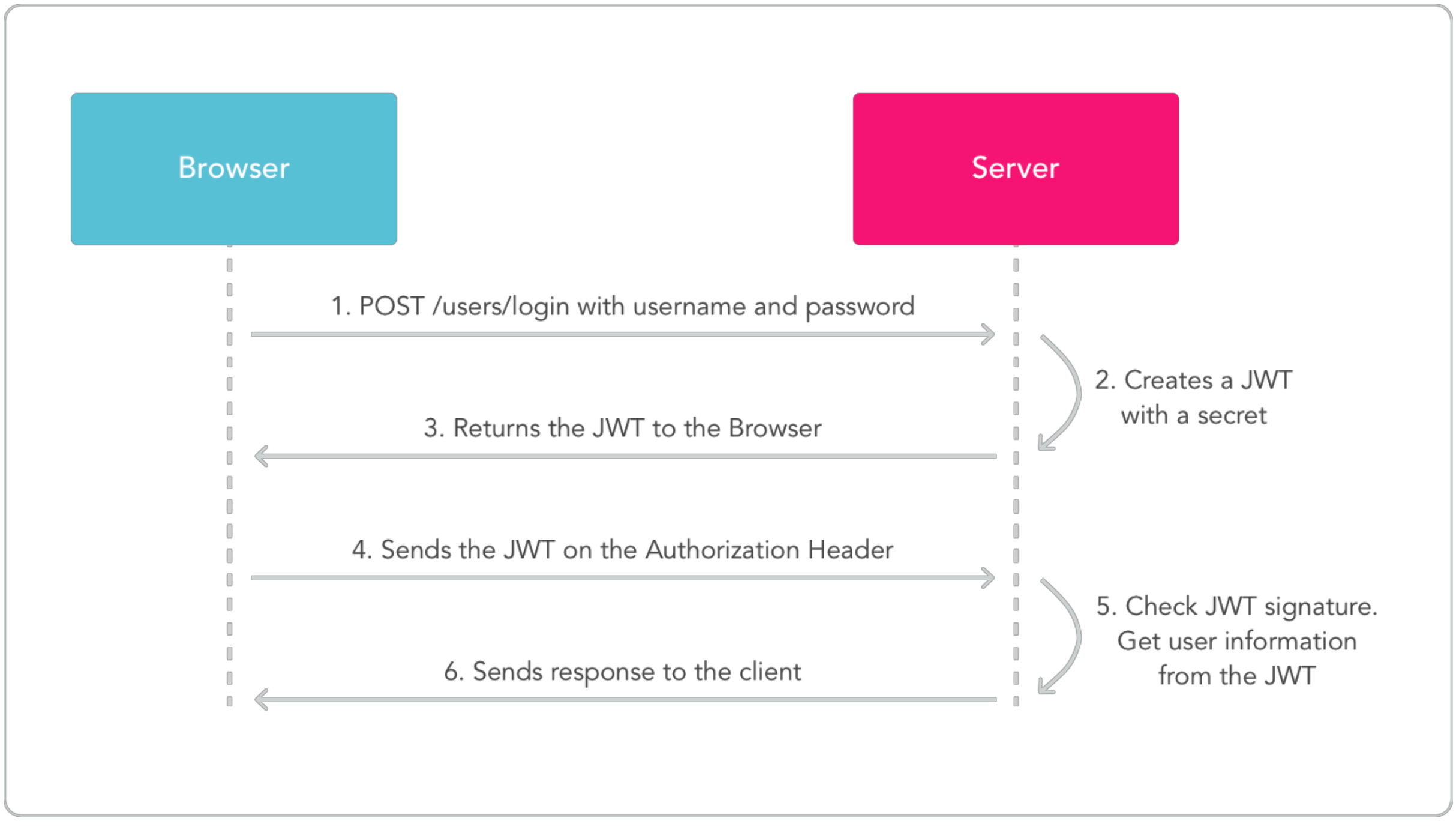
This means we will not encrypt the token

JSON web token

<header>.<payload>.<signature>

https://jwt.io/introduction/

JWT flow


Browser
Server
1. POST /users/login with username and password
2. Creates a JWT with a secret
3. Returns the JWT to the Browser
4. Sends the JWT on the Authorization Header
5. Check JWT signature. Get user information from the JWT
6. Sends response to the client

Demo

Weekly checklist

When I’m logged in

- I can see and use create, update and delete actions

When I’m not logged in

- I can only see which paths & modules exist and go through the learning process
- I can’t use the routes outside of the application without an authentication token

- Passwords are hashed when they are saved to the database

Bonus (save for the very last - these are big and difficult tasks)

- Admins have access to a user management module, in which they can add, remove and update users. Additionally, they can set user roles to either “user or “admin”
- Added users receive an email & users can reset their passwords