Authentication & Authorization

with JWT

What comes to mind when we say “Hotel Security”?

Which is Which?

Authentication

Hotel Registration

Authorization

Using our Key Card

Authentication

Hotel Registration

Login/Register

- Happens ONCE at register
- We give 2 identifiers, if they match
- We get something back (a key card)
- We get to keep the key card… until we check out
- (like Login/Register in our App)

Authorization

Using our Key Card

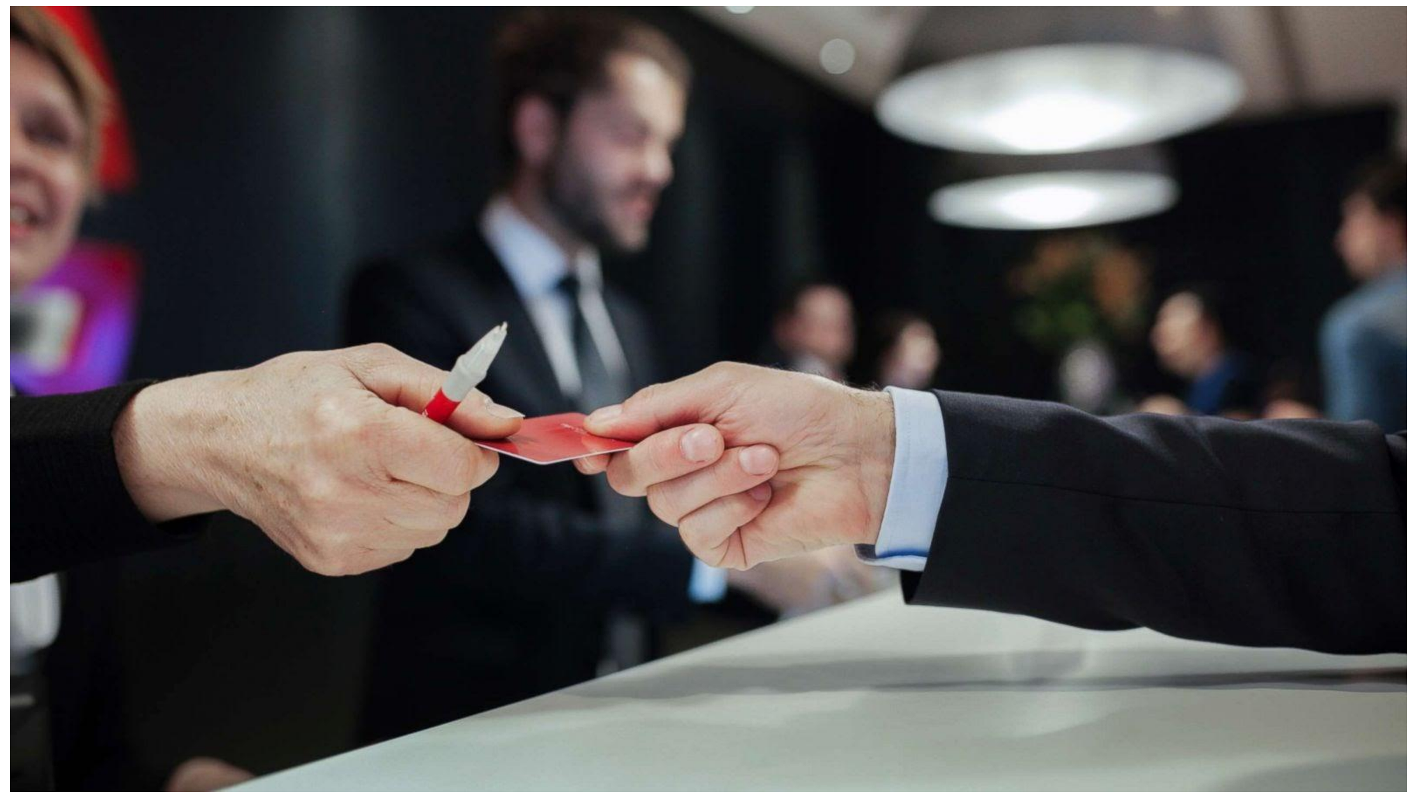
JWT - Check Token/allow access

- Reusable
- We keep it. If it gets lost, we have to “sign in again” at the front desk
- The hotel considers this substantial “proof” we are who we say we are.
- (like JWT verify in our app)

Which is Which?

Authentication	Authorization
Login/Register	JWT - Check Token/allow access
Hotel: Front Desk Registration	Hotel: Checking Key Card
Hotel: Check Name and your ID, and if it matches, give a key card	Hotel: Some fancy decryption in the lock
JWT: Check user/password. If it matches, create and send a token	JWT: Verify Token If verified, Allow user to continue to route/logic
POST api/users/login body: { username, password }	DELETE api/posts/2 headers: { Authorization: Bearer eyJhbG ciOiJIUzI1NiIsInR5cCI... }
401 Unauthorized	403 Forbidden

Authentication

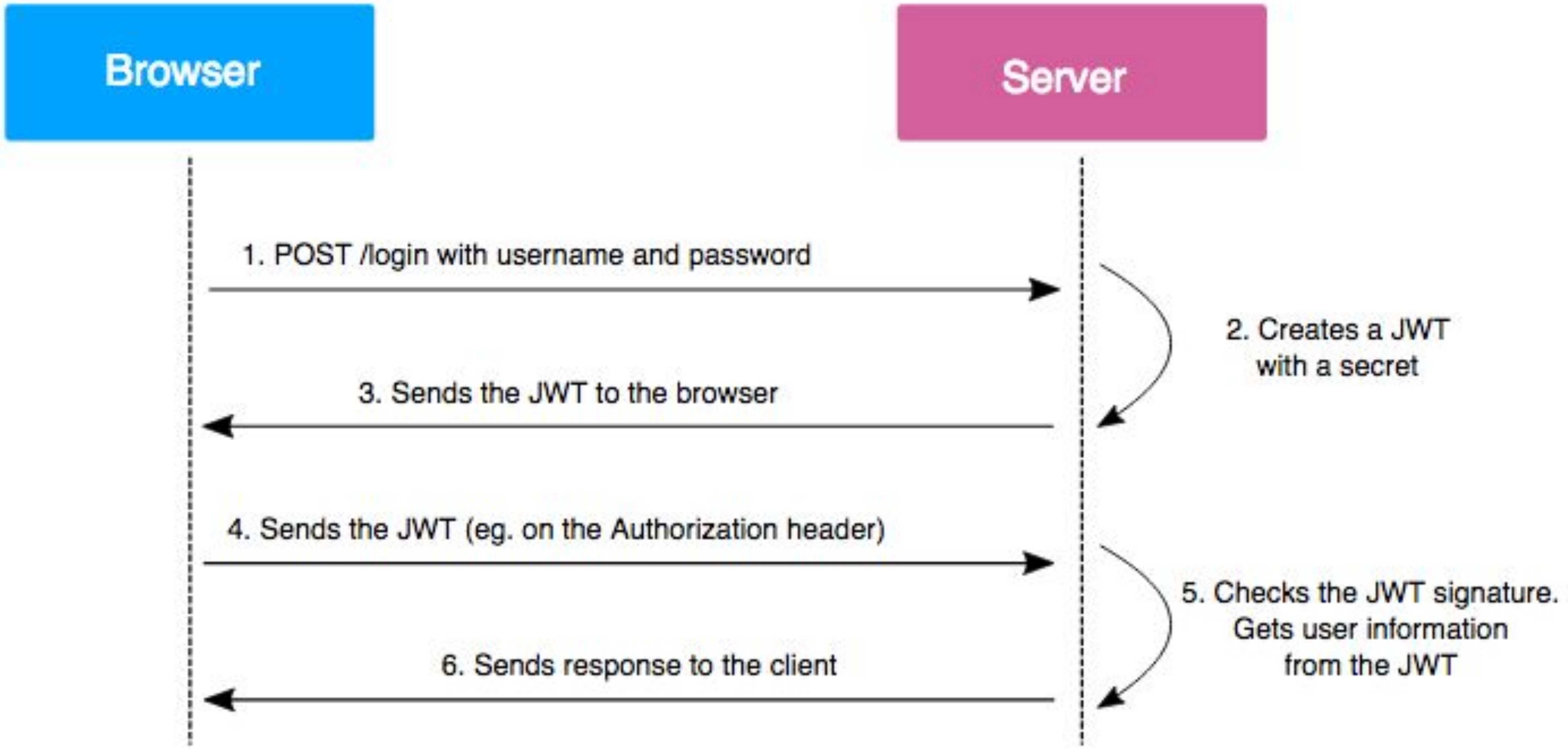


Application	Login/Register
Action	Check user/password and if it matches, then create and send back a token
Analogy	Paying for your hotel room and Creating the key card
HTTP Request Example	POST api/users/login body: { username, password }
HTTP Status Code	401 Unauthorized

Authorization

Application	JWT (verify token)
Action	Making sure the user making the request is the same user that logged in previously
Analogy	Using the key card to get into your hotel room
HTTP Request Example	DELETE api/posts/2 headers: { Authorization: Bearer eyJhbG ciOiJIUzI1NiIsInR5cCI... }
HTTP Status Code	403 Forbidden

The Cycle



1. POST /login with username and password

2. Creates a JWT with a secret

3. Sends the JWT to the browser

4. Sends the JWT (eg. on the Authorization header)

5. Checks the JWT signature. Gets user information from the JWT

6. Sends response to the client

DEMO