Page 1

Request

Response
Headers
Set-Cookie | 'ajskdp3'

Server

Some time later...

Request
Headers
Cookie | 'ajskdp3'

Payload: { userId: 'kl2j41', favoriteColor: 'red' }

JWT Creation Algorithm

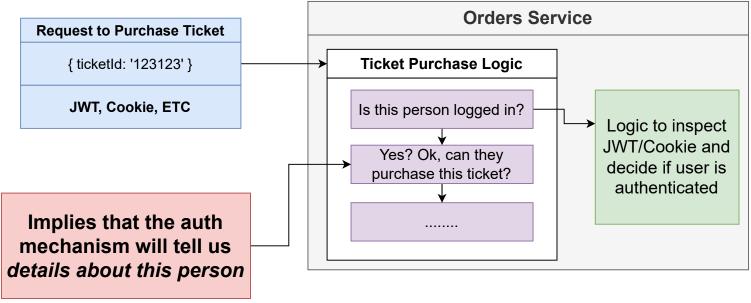JWT: 3ljl25j2lk35lk2j35.23l5k23ljk52lkj3.t34lkj63p46jp3
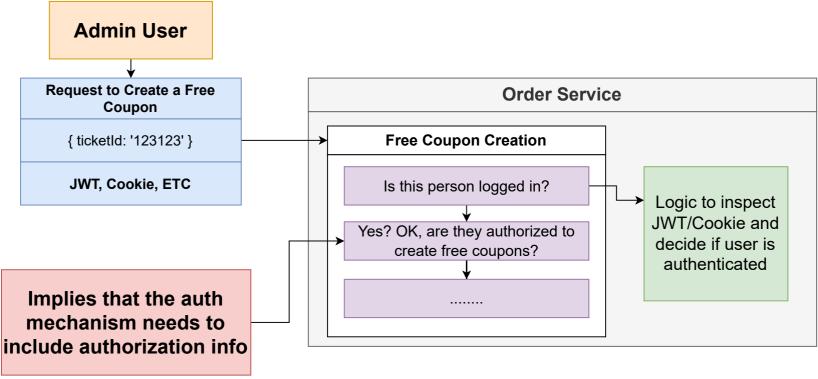
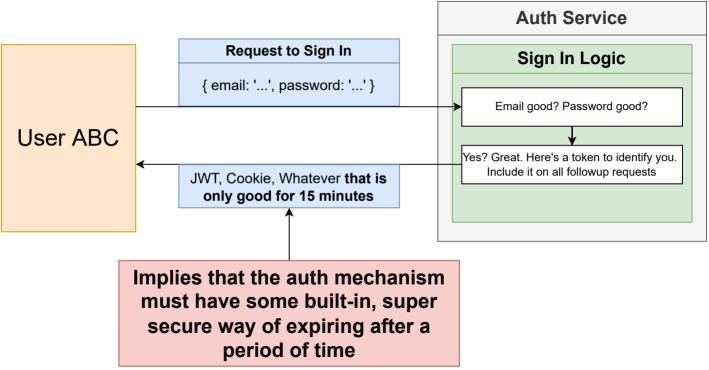| Cookies	JWT's
Transport mechanism	Authentication/Authorization mechanism
Moves any kind of data between browser and server	Stores any data we want
Automatically managed by the browser	We have to manage it manually

Request to Purchase Ticket

{ ticketId: '123123' }

JWT, Cookie, ETC

Orders Service

Ticket Purchase Logic

Is this person logged in?

Yes? Ok, can they purchase this ticket?

....

Logic to inspect JWT/Cookie and decide if user is authenticated

Implies that the auth mechanism will tell us *details about this person*

Admin User

Request to Create a Free Coupon

{ ticketId: '123123' }

JWT, Cookie, ETC

Order Service

Free Coupon Creation

Is this person logged in?

Yes? OK, are they authorized to create free coupons?

....

Logic to inspect JWT/Cookie and decide if user is authenticated

Implies that the auth mechanism needs to include authorization info

Auth Service

Sign In Logic

User ABC

Request to Sign In

{ email: '...', password: '...' }

Email good? Password good?

Yes? Great. Here's a token to identify you. Include it on all followup requests

JWT, Cookie, Whatever **that is only good for 15 minutes**

Implies that the auth mechanism must have some built-in, super secure way of expiring after a period of time

Page 1

React App

Request → **Auth Service**
Express + express-validator

Request → **Orders Service**
Ruby on Rails

Request → **Payments Service**
Java Spring

Implies that the auth mechanism must be easily understood by many languages

Implies that the auth mechanism shouldn't require some kind of backing data store on the server

Requirements for Our Auth Mechanism

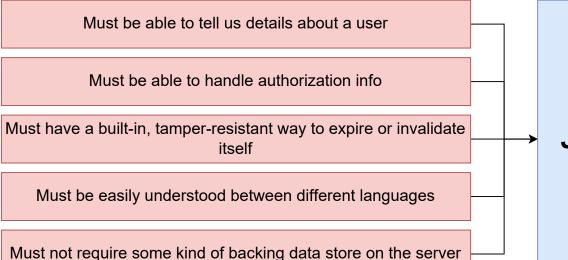Must be able to tell us details about a user

Must be able to handle authorization info

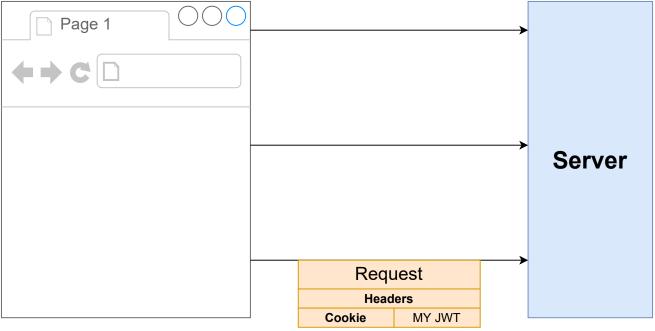Must have a built-in, tamper-resistant way to expire or invalidate itself

Must be easily understood between different languages

Must not require some kind of backing data store on the server

Requirements for Our Auth Mechanism

Must be able to tell us details about a user

Must be able to handle authorization info

Must have a built-in, tamper-resistant way to expire or invalidate itself

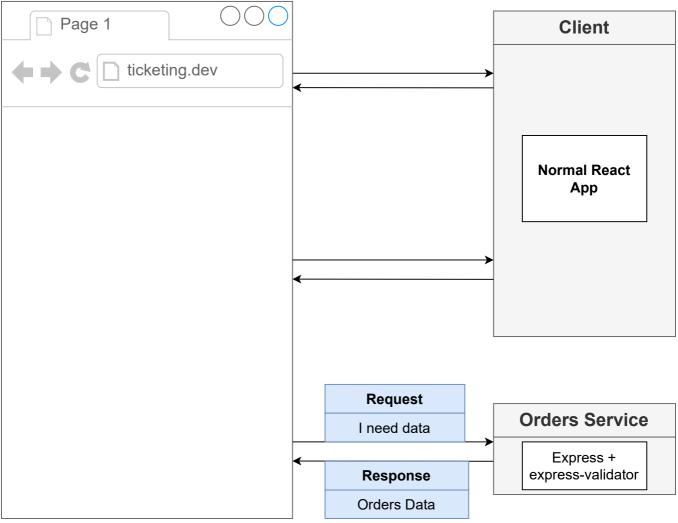| Must be easily understood between different languages |
| :-: |

Must not require some kind of backing data store on the server

JWT!

Server

Request

Headers	
Cookie	MY JWT

Page 1	
ticketing.dev	

Client

Normal React App

Request

I need data

Response

Orders Data

Orders Service

Express + express-validator

ticketing.dev

Request

GET ticketing.dev

Response

HTML file with some script tags

Request

I need JS files

Response

JS Files

Request

I need data

Response

Orders Data

Client

Normal React App

Orders Service

Express + express-validator

With a normal react app, this is the *soonest* we ever need to send auth data to the server

Page 1

GET ticketing.dev

Request

GET ticketing.dev

Response

Fully rendered HTML file with content

Client

Next JS

Orders Service

Express +
express-validator

Ticketing Service

Express +
express-validator
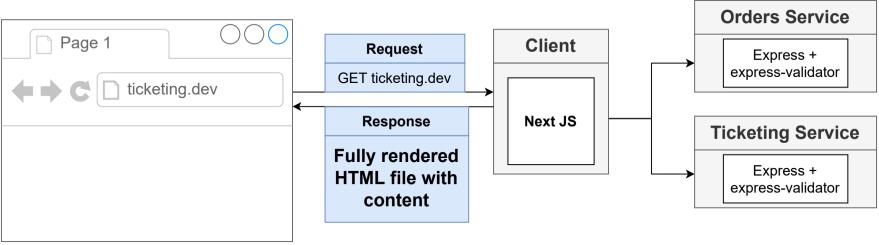
With server side rendering, we need to know auth info *with the first request*

Page 1

ticketing.dev

Request

GET ticketing.dev

Response

Fully rendered HTML file with content

Client

Next JS

Orders Service

Express + express-validator

Ticketing Service

Express + express-validator

Giant Issue!

The first request *cannot be customized in any way*

Browser window

Page 1

ticketing.dev

Request

GET ticketing.dev

Response

Fully rendered HTML file with content

Client

Next JS

Orders Service

Express + express-validator

Ticketing Service

Express + express-validator