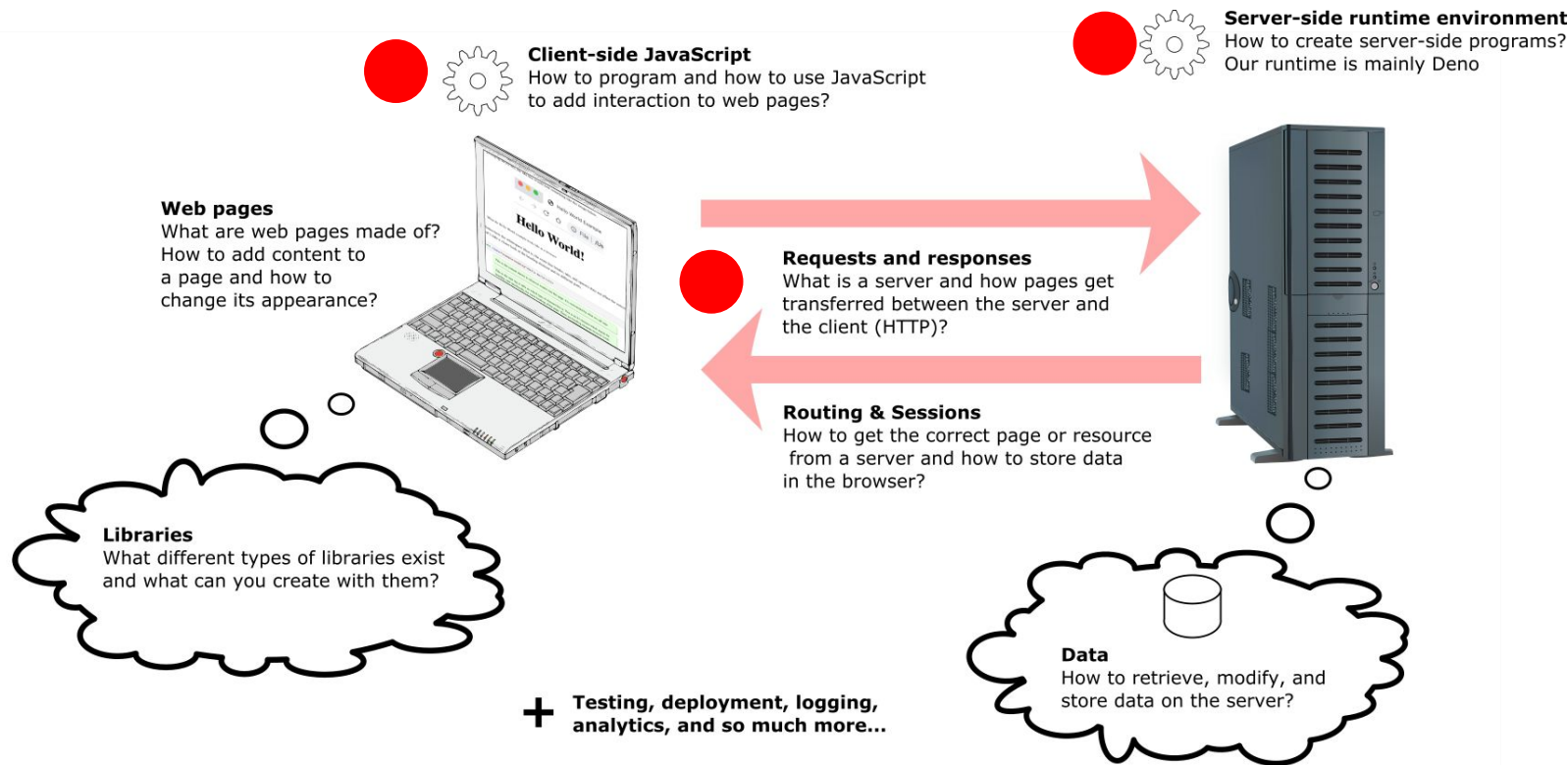


Päivä 8 - Kertaus, evästeet, autentikaatio ja API:t

2022-01-11

AaltoPRO - Websovelluskehitys

Web-sovellukset korkealla tasolla



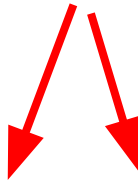
Päivä 8

- 9-12 Aamupäivä
 - Kertaus & evästeet
 - Cookies-miniprojekti
- 12:00 - 13:00 Lounas
- 13:00 - 16:00 Iltapäivä
 - Sessiot, autentikaatio, autorisaatio & käyttäjähallinta
 - Ohjelmointirajapinnat lyhyesti (API)
 - Secrets-miniprojekti
 - Seuraava "lähi"päivä

Kahvitaukoja sopivissa kohdissa

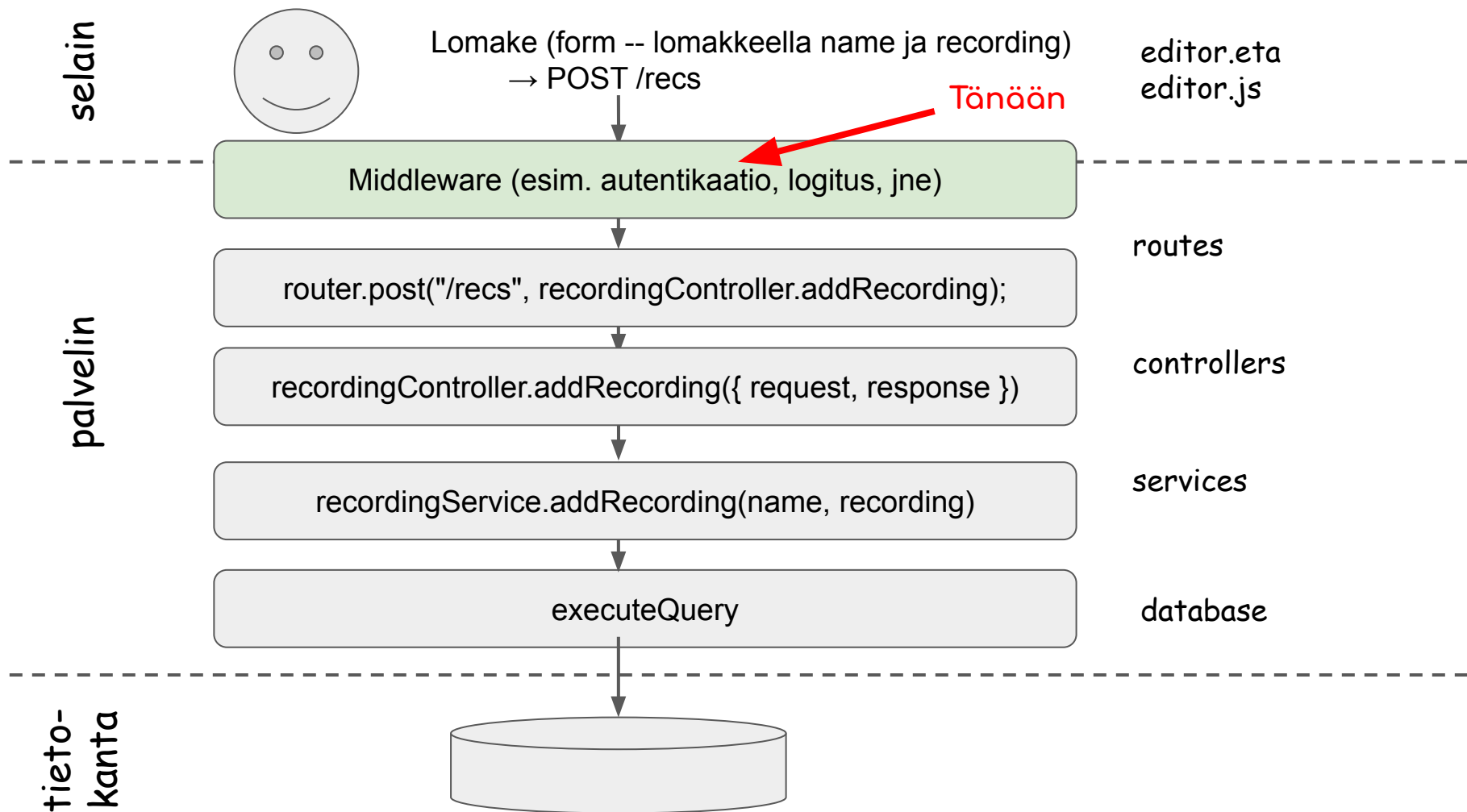
Kertausta

Oak context



```
const logout = async ({response, state}) => {
```

- Oak context [oak | A middleware framework for handling HTTP with Deno](#)
- Tänään erityisesti cookies ja state



HTTP Kertaus

Request

Request URL: http://localhost:3000/

Request Method: GET

Status Code: 🟢 200 OK

Remote Address: [::1]:3000

Referrer Policy: no-referrer-when-downgrade

Response

Response Headers [view source](#)

content-length: 531

content-type: text/html; charset=utf-8

date: Sun, 09 Jan 2022 10:16:56 GMT

set-cookie: session=_OVGywuv0hBBNx9ukw8iB; path=/; httponly

HTTP on tilaton – Kuinka säilöä tietoa pyyntöjen välillä?

- Käyttäjän selaimessa voi säilöä tietoa kolmella tavalla:
 - Evästeet (cookies)
 - Säilöt (web storage): [Web Storage API](#)
 - Local storage [Window.localStorage - Web APIs | MDN](#)
 - Session storage [Window.sessionStorage - Web APIs | MDN](#)
- Ks. lisää (erityisesti taulukko) [3 Ways To Store Data in the Browser | by Fahadul Shadhin | JavaScript in Plain English](#)

Mikä on eväste?

- Engl. cookie, browser cookie, HTTP cookie, web cookie, etc.
- Käytetään moneen tarkoitukseen
 - Session hallinta
 - Käyttäjän asetusten säilömiseen
 - Seurantaan (tracking cookies) - eikä pelkästään yhden sivuston sisällä
- MDN: [Using HTTP cookies](#)

Evästeet ominaisuudet

- Nimi ja arvo (name & value)
- Expires (MaxAge) - Kuinka kauan eväste on olemassa (ennen kuin se poistetaan)
- Domain ja Path - Onko evästeen lähettäminen sidottu johonkin domainiin ja polkuun
- SameSite [Strict, Lax, None]
 - Strict - Eväste lähetetään vain samaan domainiin (ks. yllä)
 - Lax - Eväste voidaan lähettää toiseen domainiin, mutta vain turvallisille pyynnöille (esim. GET)
 - None - Eväste voidaan lähettää kaikkiin domaineihin (esim. seurantaevästeet)
- Secure - Lähetetään vain HTTPS:n yli
- Http-only - Eväste ei saatavissa selaimessa (document.cookie)

Evästeet selaimessa

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
- http://localhost:3000

Filter

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly
keksi	arvo	localhost	/	Session	9	

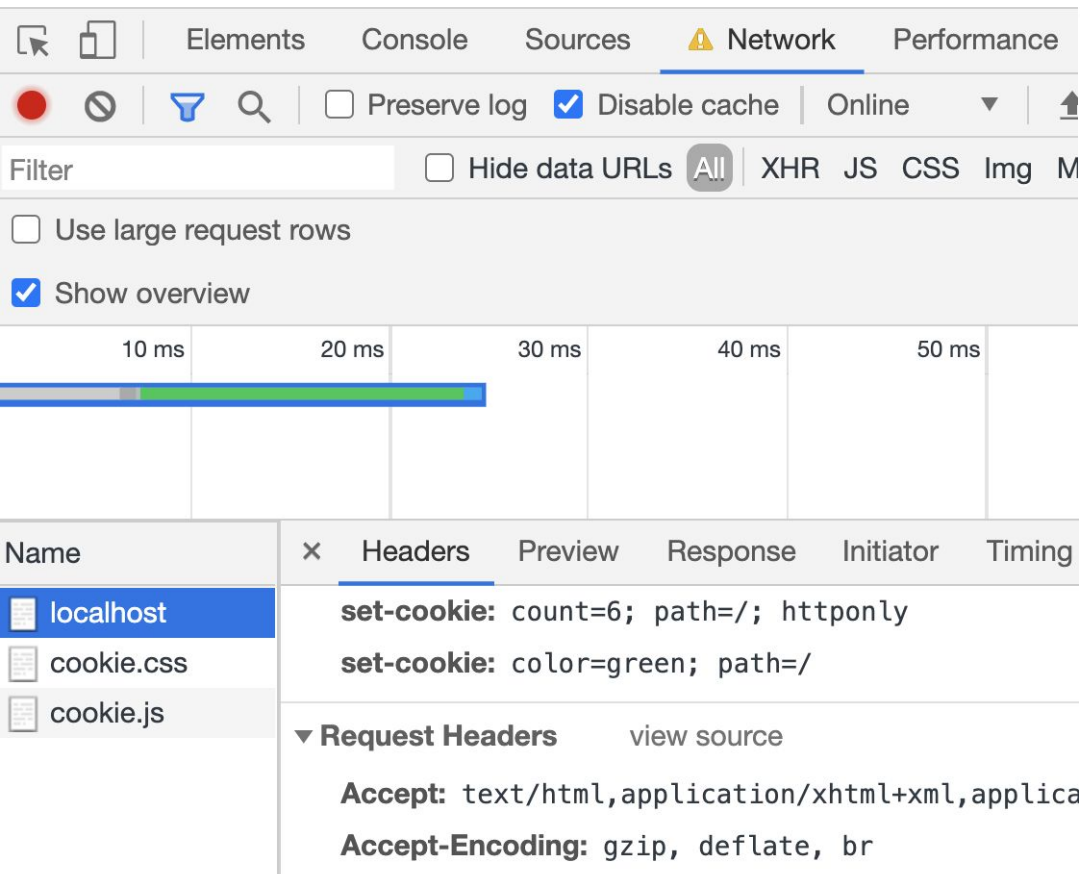
Console Sources Network Performance Memory Application Security Lighthouse Adblock Plus

Filter

☒ Only blocked

Name	Value	Domain	Path	Expire...	Size	HttpOnly	Secure	Same...	Priority
color	green	localhost	/	Session	10				Medium
count	5	localhost	/	Session	6	✓			Medium
user_name	Maija	localhost	/	Session	14	✓			Medium

Evästeet selaimessa



Elements Console Sources **Network** Performance

Filter ☐ Hide data URLs **All** XHR JS CSS Img M

☐ Use large request rows

☒ Show overview

10 ms 20 ms 30 ms 40 ms 50 ms

Name	Headers	Preview	Response	Initiator	Timing
localhost	set-cookie: count=6; path=/; httponly				
cookie.css	set-cookie: color=green; path=				
cookie.js	Request Headers view source				
	Accept: text/html,application/xhtml+xml,application/javascript;q=0.9,*/*;q=0.8				
	Accept-Encoding: gzip, deflate, br				

▼ Response Headers

[view source](#)

content-length: 922

content-type: text/html; charset=utf-8

date: Sun, 09 Jan 2022 07:23:11 GMT

set-cookie: count=6; path=/; httponly

set-cookie: color=green; path=

document.cookie

[Document.cookie - Web APIs | MDN](#)

▼ Response Headers

[view source](#)

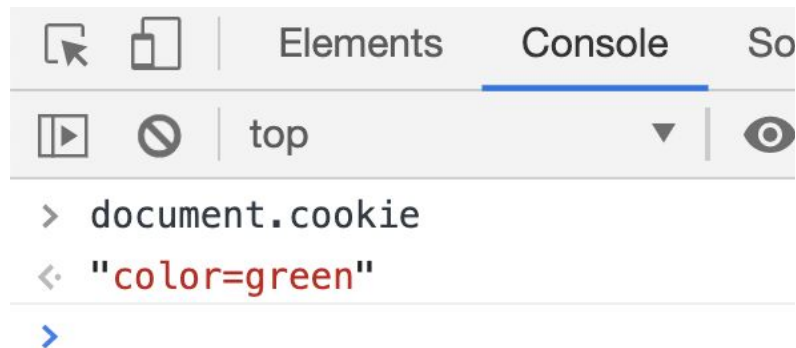
content-length: 922

content-type: text/html; charset=utf-8

date: Sun, 09 Jan 2022 07:23:11 GMT

set-cookie: count=6; path=/; **httponly**

set-cookie: color=green; path=/



Kolmannen osapuolen evästeet (3rd party cookies)

- Kolmannen osapuolen, eli eri domainiin kuuluvat evästeet mahdollistavat esimerkiksi käyttäjän seuraamisen ja sessiot eri domaineissa
- Voidaan käyttää hyvässä ja pahassa
- GDPR ja muut lait säätelevät näiden käyttöä

Evästeet ja Deno & Oak

<https://deno.land/x/oak@v10.1.0/cookies.ts#L19>

Asettaminen:

Arvo

Asetukset

```
const index = async ({ cookies, response }) => {  
  await cookies.set("keksi", "arvo", {httpOnly: false});  
  response.body = await renderFile("../views/index.eta");  
}
```

Nimi

Lukeminen:

```
const index = async ({ cookies, response }) => {  
  console.log(await cookies.get("keksi"))  
  response.body = await renderFile("../views/index.eta");  
}
```

Miniprojekti - Cookie saver

<https://github.com/aaltopro-weblearners/project-09a-cookies>

1. Toteuta uuden käyttäjän lisääminen (kertausta)
 - a. routes/controllers/cookieController.js : addUser -> käyttäjän nimi ja lempiväri
 - b. services/userService.js : saveNewUser -> Tallentaa käyttäjän tietokantaan
2. Toteuta evästeen tallennus käyttäjän syötteestä
 - a. routes/controllers/cookieController.js : saveCookie -> Lisää uuden evästeen käyttäjän antamalla arvoilla
3. Toteuta latauslaskuri (vinkki: <https://wsd.cs.aalto.fi/14-cookies-and-sessions/2-cookies/>)
 - a. routes/controllers/cookieController.js : index ->
 - i. Jos eväste "counter" on olemassa, inkrementoi laskurin yhdellä ja tallentaa sen evästeeseen...
 - ii. ...jos ei, luo evästeen "counter" ja asettaa sen arvoksi 1
4. Toteuta käyttäjän lempiväriin lisääminen evästeeseen "color"
 - a. services/userService.js : getUserColor -> palauttaa lempiväriin merkkijonona käyttäjänimen perusteella
 - b. routes/controllers/cookieController.js : index -> Asettaa "color" evästeen getUserColor-palvelun avulla
 - c. httpOnly - Jos color-eväste ei ole http-only päällä, vaihtuu laatikon väri selaimessa (selaimen js toteutettu valmiiksi)

Lounas

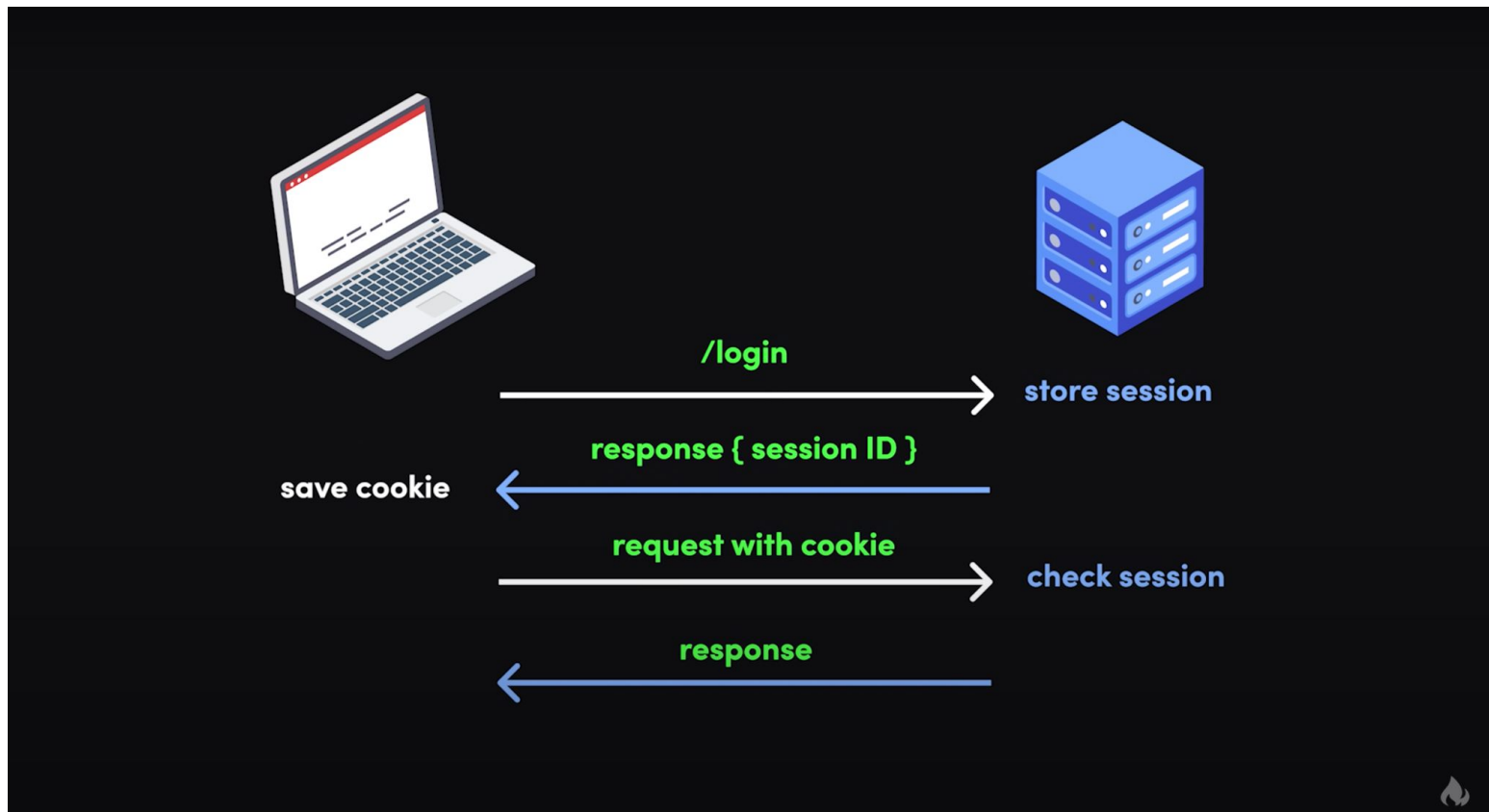
Autentikaatio ja Autorisaatio

- Autentikaatio (authentication)
 - Kuka olet?
 - Jos ei tiedossa/autentikaatio epäonnistui HTTP 401 [401 Unauthorized - HTTP | MDN](#)
- Autorisaatio (authorization)
 - Onko sinulla pääsyä resurssiin
 - Jos resurssi ei sallittu (tälle käyttäjälle) HTTP 403 [403 Forbidden - HTTP | MDN](#)
- Esim. tavallinen käyttäjä voi kirjautu sivustolle, mutta hänellä ei ole oikeutta nähdä muiden käyttäjien tietoja

Sessiot

- Sessiot ovat tapa pitää yllä tilaa pyyntöjen välillä (esim. ostoskorin sisältö, käyttäjän sisäänkirjautuminen)
- Vaihtoehtoja
 - Sessio-evästeet
 - Tokenit
- JSON Web Token
 - [JSON Web Token - Wikipedia](#)

Sessio evästeillä



Sessio evästeillä



1. User submits login form



SESSION ID
←



2. Server stores a session



3. Browser puts session ID in cookies



→
←



4. Browser sends cookies with future requests

Session
oikeellisuus
tarkistetaan
tietokannasta

Token-autentikaatio



1. User submits login form



2. Server creates a JWT
(JSON Web Token)



3. Browser puts JWT in local storage



Authorization: Bearer <token>



4. Signed JWT header validated
on future requests

Ei vaadi
Tietokanta-
kyselyä

Ohjelmointirajapinnat (Application Programming Interface (API))

- Suunnittelu, käyttö
- Representational State Transfer (REST)
- Restful
 - Tilaton
 - Jokaisella resurssilla oma uniikki URI (uniform resource identifier), esim.
 - `https://esimerkki.com/api/v1/vihannekset` ← Palauttaa listan vihanneksista
 - `https://esimerkki.com/api/v1/vihannekset/7` ← Palauttaa yksittäisen resurssin, esim.





```
{  
  "id": 7,  
  "nimi": "porkkana"  
}
```
 - Useimmiten käytetään JSONia
 - Monesti token-autentikaatio

Sessiot ja Deno & Oak

```
await state.session.set("authenticated", true);
await state.session.set("user", {
  id: userObj.id,
  name: userObj.name,
  admin: userObj.admin,
});
```

```
const logout = async ({response, state}) => {
  await state.session.set("authenticated", false);
  response.redirect("/");
}
```

Sessiot ja Deno & Oak

Console Sources  Network Performance Memory Application Security Lighthouse						
Filter    <input type="checkbox"/> Only blocked						
Name	Value	Domain	Path	Expires ...	Size	HttpO...
session	_OVGywuvOhBBNx9ukw8iB	localhost	/	Session	28	✓

Huom.



```
> document.cookie
```

```
< ""
```

```
>
```


Salasanojen tallentaminen kantaan

- Ei koskaan salasanoja kryptaamattomana tietokantaan
- BCrypt [bcrypt@v0.2.4 | Deno](#)

```
const password = params.get("password");
```

```
const passwordCorrect = await bcrypt.compare(password, hash);
```

- Kryptaus ei suojaakaan, jos salasana heikko
 - [Wikipedia:10,000 most common passwords](#)
 - [Rainbow table - Wikipedia](#)

bcrypt

```
// deno run --unstable --allow-net app.js
import * as bcrypt from "https://deno.land/x/bcrypt@v0.2.4/mod.ts";

const hash = await bcrypt.hash("asparagus");

console.log('Comparing hash with password');
let result = await bcrypt.compare("password", hash);
console.log(`Were they the same? ${result}`);

console.log('Comparing hash with asparagus');
result = await bcrypt.compare("asparagus", hash);
console.log(`Were they the same? ${result}`);

console.log(hash)
console.log(bcrypt.hash("password")) //bcrypt.hash returns a promise
console.log(await bcrypt.hash("password"))
```

bcrypt - mutta!

```
// deno run --unstable --allow-net app.js
import * as bcrypt from
"https://deno.land/x/bcrypt@v0.2.4/mod.ts";

let hash = await bcrypt.hash("asparagus");
console.log(hash);
hash = await bcrypt.hash("asparagus");
console.log(hash);

// mitä ihmettä!
```

Deno & HTTP status

```
if (!passwordCorrect) {  
  response.status = 401;  
  response.body = "Unauthorized";  
  return;  
}
```

Miniprojekti - Secrets

- <https://github.com/aaltopro-weblearners/project-09b-secrets>
- Mukautettu projekti, paljon lisäapua:
<https://wsd.cs.aalto.fi/15-authentication-and-authorization/2-credentials-in-database/>
- Valmiit käyttäjät
 - user/user
 - admin/admin

Miniprojekti - Secrets - Tehtävää 1/2

- The application has already the following functionality:
- GET / redirects the user to the login form or shows content, depending on whether the user has authenticated.
- GET /auth/register shows a registration form.
- POST /auth/register registers the user and redirects the user to /auth/login.
- GET /auth/login shows a login form.
- Your task is to implement the authentication functionality
(`routes/controllers/authenticationController.js : postLoginForm`). The functionality should work as follows. When the client makes a POST request to /auth/login, the server should:
 - (1) verify that a user with the given email address exists in the database -- if not, the server should respond with the status code 401;
 - (2) verify that the password for the user with the corresponding email matches the password hash stored in the database (use compare-function of bcrypt-library) -- if not, the server should respond with the status code 401;
 - (3) if the previous checks pass, set the session key "authenticated" as true and add an object "user" to the session -- the "user" object in the session should have the id and the email of the authenticated user;
 - (4) finally, when the user has authenticated, the user should be redirected to the path "/".

Miniprojekti - Secrets - Tehtävää 2/2

- Mahdollista käyttäjälle salaisuuden vaihtaminen (huom. autentikaatio!)
- /admin - Kaikkien käyttäjien ja salaisuuksien listaaminen
 - Jos käyttäjä on kirjautunut sisään JA on admin oikeudet, näytetään listaus
 - Jos käyttäjä on kirjautunut sisään, mutta EI ole admin-oikeuksia näytetään 403 Forbidden
 - Jos käyttäjä ei ole kirjautunut sisään, näytetään 401 Unauthorized
- Linkki /admin -resurssiin näytetään etusivulla ainoastaan, jos käyttäjällä on admin-oikeudet
- Bonus-tehtäviä:
 - Anna mahdollisuus "admin" -roolilla varustetulle henkilölle antaa muille käyttäjille admin rooli
 - Luo uusi käyttäjärooli "moderator", joka voi nähdä käyttäjälistauksen, mutta ei voi antaa admin-oikeuksia kenellekään (ei myöskään itselle). Anna admin-henkilöille mahdollisuus antaa moderator-rooli käyttäjille
 - Hyvä tapa toteuttaa erilaisia käyttäjärooleja:
<https://wsd.cs.aalto.fi/15-authentication-and-authorization/3-authorization/#role-based-access-control>

Seuraava "lähi"päivä

- Toistaiseksi sisältö on aina tuotettu palvelimelle
- Seuraavana lähipäivänä:
 - APlen kertausta
 - Selainpuolen toiminnallisuutta