

Introduksjon til webutvikling

20. august 2018

Dag 2

Agenda

09.00: Gjennomgang av nedteller

09.15 (ish): Nettverkskall og JavaScript

Jobbe med oppgave om bildegalleri

11.00 (ish): Media queries og responsivt design

Fortsette på oppgave om bildegalleri

12.00: Lunsj

Jobbe med oppgave om boligannonser

15.00: Prodsett din første app!

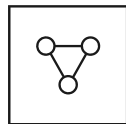
Jobbe med det man selv vil

16.30: Avslutning

JavaScript

Litt mer avanserte konsepter, fortsatt ganske raskt forklart

SPA

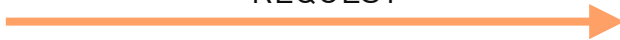


Single Page Application

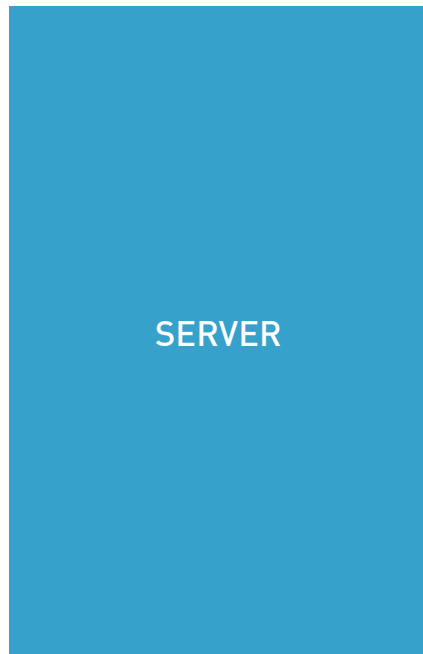
Laster innhold dynamisk, i stedet for full sidelast ved hver navigasjon



REQUEST



SERVER

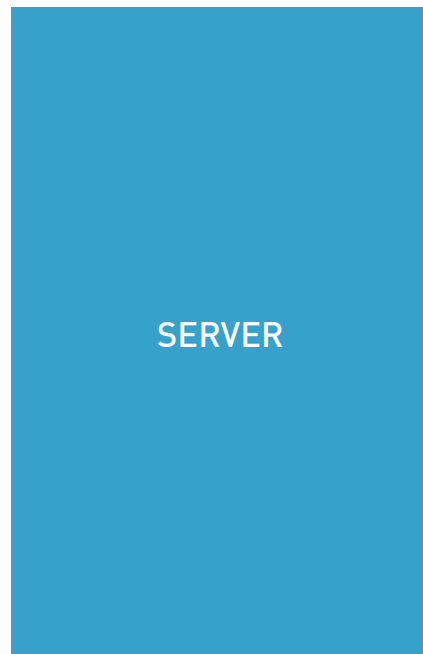
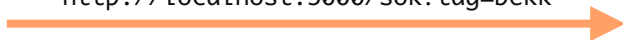


RESPONSE



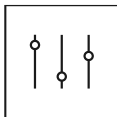


`http://localhost:5000/sok?tag=bekk`



```
[  
  {  
    "url": "https://flickr.com/bilde1.jpg",  
    "title": "Fint bilde av en bekker"  
  },  
  {...}  
]
```

API



Application programming interface

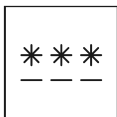
Grensesnitt av ymse sort, feks.
predefinerte funksjoner i et programmeringsspråk,
eller grensesnitt inn mot web-servere

REST



Representational state transfer

Arkitekturstil som definerer et sett med begrensninger ved oppsett av web-tjenester (*web services*)



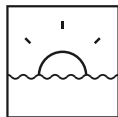
RESTful web services

Endepunkter som følger REST-stilen. Vi støter som oftest på operasjonene GET, POST, PUT og DELETE

Laste inn data

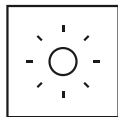
```
fetch('http://localhost:5000/sok?tag=bekk');
```

Asynkronitet



Gamlemåten: Promises

Chaining med resolve og reject



Den nye og bedre måten: async/await

Men fungerer ikke i IE :(

Laste inn data

```
async function hentData() {  
  const dataPromise = fetch('http://localhost:5000/sok?tag=bekk');  
  return await dataPromise;  
}
```

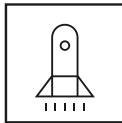
OBS!



Vi kan bare kalle await i async-funksjoner

Vil ikke fungere på toppnivå

JSON



JavaScript Object Notation

Brukes til å sende komplekse data over nettverk

JSON

```
{  
  "string": "en string",  
  "key2": 1.2,  
  "boolean": false,  
  "array": [1,2,3,4,5,6],  
  "object": {  
    "key": "value"  
  },  
  "empty": null  
}
```

JavaScript-objekt

```
{  
  string: "en string",  
  key2: 1.2,  
  boolean: false,  
  array: [1,2,3,4,5,6],  
  object: {  
    key: "value"  
  },  
  empty: null  
}
```

Parse JSON

```
JSON.parse('{"key":"value"}'); => { key: «value»}
```

```
JSON.stringify({key:"value"}); => { "key": "value"}
```

Parse fetch-response

...til JSON

```
const response = await fetch('https://ressurs.no');  
return await response.json();
```

...til tekst

```
const response = await fetch('https://ressurs.no');  
return await response.text();
```




Alt sammen samtidig takk!

fetch + parsing + feilhåndtering

Laste inn data

...nærmer oss nå

```
async function hentData() {  
  const dataPromise = fetch('http://localhost:5000/sok?tag=bekk');  
  const dataResponse = await dataPromise;  
  const jsonPromise = dataResponse.json();  
  const json = await jsonPromise;  
  return json;  
}
```

Komprimert

```
async function hentData() {  
  const data = await fetch('http://localhost:5000/sok?tag=bekk');  
  return await data.json();  
}
```

Med feilhåndtering

```
async function hentData() {  
  try {  
    const data = await fetch('http://localhost:5000/sok?tag=bekk');  
    return await data.json();  
  } catch (err) {  
    // gjør noe med feilen  
  }  
}
```

Med Promise

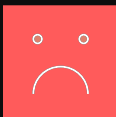
```
function hentData() {  
  fetch('http://localhost:5000/sok?tag=bekk')  
    .then(function(response) {  
      return response.json();  
    })  
    .then(function(data) {  
      return data;  
    })  
    .catch(function(error) {  
      // gjør noe med feilen  
    });  
}
```



Alle asynkrone funksjoner returnerer et Promise

Uavhengig av om funksjonen er implementert med
.then og .catch eller async/await

```
fetch('http://ntnu.no');
```



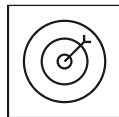
XMLHttpRequest cannot load <https://ntnu.no>.
No 'Access-Control-Allow-Origin' header is present on the requested resource.
Origin <http://localhost:5000> is therefore not allowed access.

Same origin policy



Vi kan ikke få tilgang på hvilket som helst endepunkt

Fordi sikkerhet

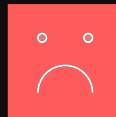


Origin

Kombinasjon av URI-skjema, hostnavn
og portnummer

`http://localhost:5000`

`https://ntnu.no:443`

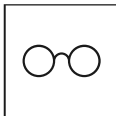


`http://localhost:5000`

`http://localhost:5000/sok?tag=bekk`



Oppgavetid!



GitBook

Oppgave: bildegalleri



Lurer du på hvorfor noe er som det er eller sitter du fast?

Spør oss :D Om alt! *Ingen spørsmål er dumme*



Føler du at du har forstått greia?

HOPP OVER DET! :D