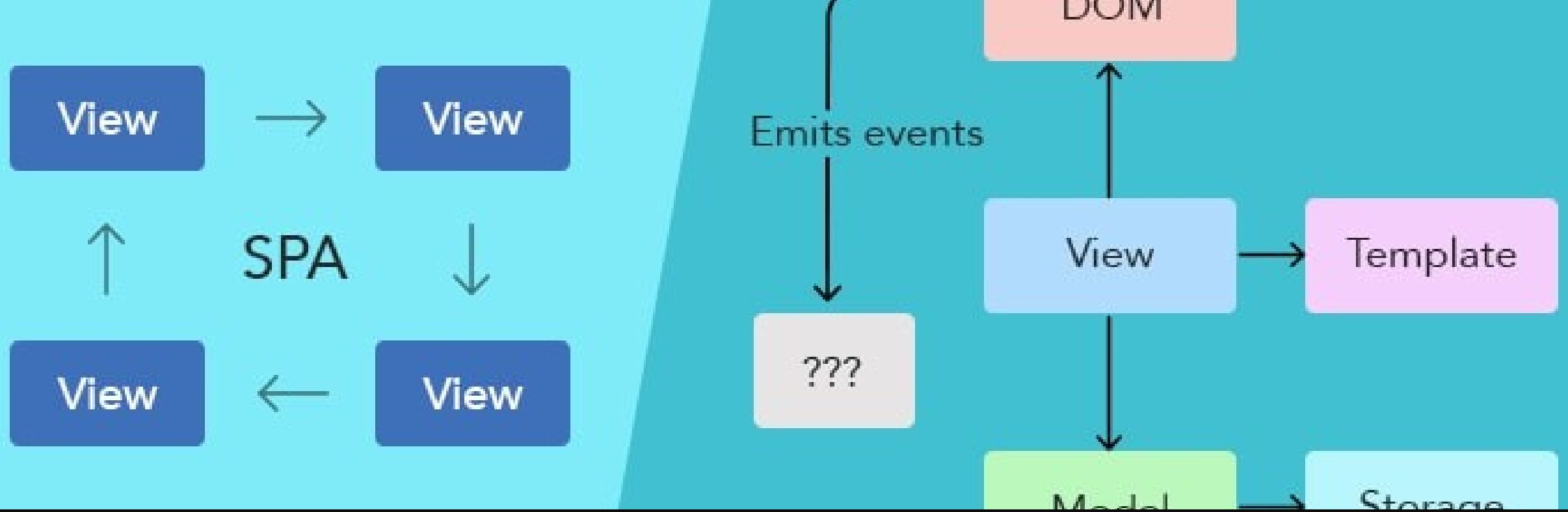


# GUI Programming & Single-page App

Simon Hoxer Bønding

Faglærer

E-mail: [sinb@techcollege.dk](mailto:sinb@techcollege.dk)



# Single-Page Application introduktion

# Målepindene

- Eleven kan beskrive arkitekturen for en Single Page Application (SPA) med dens fordele og ulemper.
- Eleven kan udvikle en SPA client vha. et eller flere SPA frameworks.
- Eleven kan anvende og konfigurere komponenter, moduler, services osv. i et eller flere frameworks.
- Eleven kan udvikle en SPA client med flere "sider" vha. Routing.
- Eleven kan udvikle en SPA client, der vha. HTTP kommunikerer asynkront med en webservice.
- Eleven kan redegøre for sikkerhedsmæssige udfordringer ved en SPA løsning.
- Eleven kan debugge en SPA client.

## Lektion 1 - Introduktion

1. **Målepindene**
2. **Hvordan vi arbejder**
3. **Repetition af clientside**

# Flere muligheder

- Hvis du har tidligere erfaring med Vue/React etc.
  - Find guide eller lav egen opgave
- Hvis du gerne vil starte med basics
  - Pluralsight i moodle om HTML, CSS, javascript
  - Youtube i moodle med Vue3 kursus

## Lektion 1 - Introduktion

1. Målepindene
2. **Hvordan vi arbejder**
3. Repetition af clientside



**DATAKOMMUNIKATION**  
EN UDDANNELSE PÅ **TECH**COLLEGE

# Repetition af clientside

1. Hvordan laver man et link i HTML?
2. En HTML tabel?
3. Et billede?
4. Hvad rammer selectoren: div.title
5. Hvad rammer selectoren: #heading
6. Hvordan ser HTML grundstrukturen ud?
7. Hvilke typer indeholder javascript?
8. Skriv via javascript "Hej verden" i html'en i `<div id="content"></content>`

## Lektion 1 - Introduktion

1. Målepindene
2. Hvordan vi arbejder
- 3. Repetition af  
clientside**



**DATAKOMMUNIKATION**  
EN UDDANNELSE PÅ **TECH**COLLEGE

# Gængse attributter

Attribut		
id	Unik identifikator for et HTML-element.	<div id="header">Indhold</div>
class	Angiver en eller flere klasser for et element.	<p class="tekst stor">Hej</p>
style	Indsætter inline CSS-styling for et element.	<h1 style="color: red;">Overskrift</h1>
src	Angiver kilden til et billede eller mediefil.	
href	Angiver URL'en for et hyperlink.	<a href="https://www.eksempel.dk">Link</a>
alt	Beskrivelse af et billede, hvis det ikke kan vises.	
title	Viser en tooltip-besked ved mouseover.	<span title="Dette er en tooltip">Hover over mig</span>
target	Bestemmer, hvor et hyperlink åbnes.	<a href="https://www.eksempel.dk" target="_blank">Åbn i ny fane</a>
name	Angiver navnet på et element, ofte brugt i formularer.	<input type="text" name="brugernavn">
value	Angiver default værdien for inputfelter.	<input type="text" value="Hej">

## Lektion 1 - Introduktion

1. Målepindene
2. Hvordan vi arbejder
- 3. Repetition af  
clientside**

# Single-page application definition

- En SPA er en webapplikation eller webside, som interagerer med webbrowseren ved dynamisk at omskrive den nuværende side med nye data fra webserveren.
  - Dvs. én primær forespørgsel til web serveren. Derefter bruges javascript til at hente enten JSON, XML eller brudstykker af HTML til at omskrive serveren.

## Lektion 2 – Single-page applikationen

1. **Single-page application definition**
2. Det helt store **hvorfor?**
3. Javascript programmer?
4. Opgave
5. Fordele og ulempler ved SPA



# Det helt store **hvorfor!**?

- Hvorfor beskæftige sig med SPA?
- SPA ligner mere et "rigtigt" program
  - Mindre afhængig af serveren
  - Overlader mere af applikationens kodelogik til klienten.
  - Føles mere smooth af brugeren
  - Lettere at omskrive til webapp til devices
  - Lettere at omskrive til offline brug

## Lektion 2 – Single-page applikationen

1. Single-page application definition
2. **Det helt store hvorfor?**
3. Javascript programmer?
4. Opgave
5. Fordele og ulempler ved SPA





# Javascript programmer!?

## ER I %α" # VANVITTIGE!!!

- Javascript har modnet sig
  - Alle mulige devices
  - Frameworks (jQuery, Angular)
  - Dynamik (Live ændring DOM-træ)
  - Udbredelse (Websites, Node.JS, inDesign)
- Single-page applications er en nævneværdig mulighed til mindre programmer og SAAS

## Lektion 2 – Single-page applikationen

1. Single-page application definition
2. Det helt store **hvorfor?**
3. **Javascript programmer?**
4. Opgave
5. Fordele og ulempler ved SPA



# Opgave

1. 2-2 find et eksempel på en Single-page application
2. Yeet den ind i denne padlet <http://kortlink.dk/28vvr>
3. Tilføj et par sætninger, der beskriver den SPA I valgte!

# Fordele og ulemper ved SPA

## Fordele

- Load balancing (Klienter tager en del af arbejdet)
- Optimering (Overflødig HTML data sorteres væk)
- Lettere konvertering til PWA
- Et skridt væk fra sessionstyret dataadgang

## Bagdele

- Søgemaskineoptimering
- Serverlogik i clientside template
- Browser historik
- Analytics
- Sikkerhed

# Phew! Det var lektion 2 – Single-page application

Nu skal vi til lektion 3: Webarkitektur

Hvor meget fylder serveren?

Hvordan opfører HTTP sig?

Kan vi lave en webserver?

# Webarkitekturen

Njjaauwn

# Generelt om arkitektur

## Thick stateful server architecture

- Webserveren husker på klientens tilstand (hvor han på siden)



## Thick stateless server architecture

- Webserveren "udregner" hvor klienten er, baseret på noget klienten siger

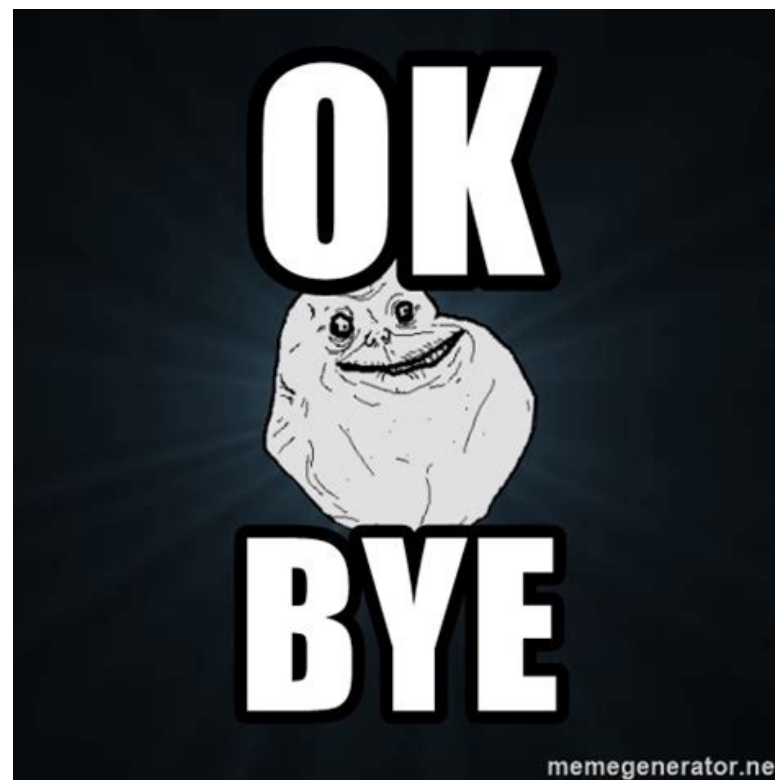


# Generelt om arkitektur

## Thin server architecture

- Webserveren sender HTML/CSS og javascript filer til klienten, og ser ikke meget til klienten derefter.

## Webserveren:





# Er det thick stateful, thick stateless eller thin arkitektur?

- Moodle
- Facebook
- Dr.dk
- Gmail
- Discord



# Her er en HTML tabel

Hvor dannes  
HTML'en?

**Thick stateful server  
architecture:** Server  
laver HELE siden

**Thick stateless  
server architecture:**  
Server laver kun  
tabellen

**Thin server  
architecture:** Klient  
laver tabellen.

Elev	Fag	Karakter
Anne Dam	GUI-programmering	02
Steen Secher	GUI-programmering	12
Ann Tenna	GUI-programmering	NaN
Remo Lademann	GUI-programmering	4
Ella Fanth	GUI-programmering	10
Steen Isaksen	GUI-programmering	00
Ejvind Møller	GUI-Programmering	02
Side 1		Næste side

# Recap

- SPA flytter logik (særligt GUI relateret logik) til klienten
  - F.eks. Skiftende baggrundsfarve i et grid, visning af en loginform, first-hand validering
- Der er fordele og ulemper ved SPA
- SPA er måske en mere "rigtigt" løsning på web



# Simpel webserver

```
class SimpleHttpServer
{
    public static async Task Main(string[] args)
    {
        using (HttpListener listener = new HttpListener())
        {
            listener.Prefixes.Add("http://localhost:8080/");
            listener.Start();
            Console.WriteLine("Server is listening on http://localhost:8080/");

            while (true)
            {
                // Skaf request context
                HttpListenerContext context = await listener.GetContextAsync();
                HttpListenerRequest request = context.Request;
                HttpListenerResponse response = context.Response;
                // set response type til html
                response.ContentType = "text/html";

                string responseHtml = "<html><body><h1>Hello, World!</h1><p>Simpel webserver i C#.</p></body></html>";

                // Convert the response string to a byte array
                byte[] buffer = Encoding.UTF8.GetBytes(responseHtml);

                // Write the response
                response.ContentLength64 = buffer.Length;
                await response.OutputStream.WriteAsync(buffer, 0, buffer.Length);

                // Close the response
                response.Close();
            }
        }
    }
}
```

# Vue.js

Udtales "view", men sig bare "vy" for at drille

# Hvad minder vue.js om?

- Vue.js er skrevet i Javascript og minder om React og Angular
- WPF og komponenttankegangen i Blazor
- Tilbyder databinding og templating
  - Databinding: Ændringer i data afspejles i viewet.
  - Templating: Smart når kode skal gentages mange gange.
- Meget modulært

1. **Hvad minder vue.js om?**
2. Hvordan ser det ud?
3. Demonstration
4. Opgave

# Hvordan ser det ud?

- En Vue-instans varetager strukturen under et bestemt HTML tag
- Instansen indeholder
  - Konfiguration af instansen
  - Variabler som bruges i HTML
  - Skabelon
- Lad os se en video

1. Hvad minder vue.js om?
- 2. Hvordan ser det ud?**
3. Demonstration
4. Opgave

# Igang det går

Afprøv Vue.js på egen hånd. Læs kom-i-gang guiden på  
vuejs.org

<https://vuejs.org/v2/guide/>

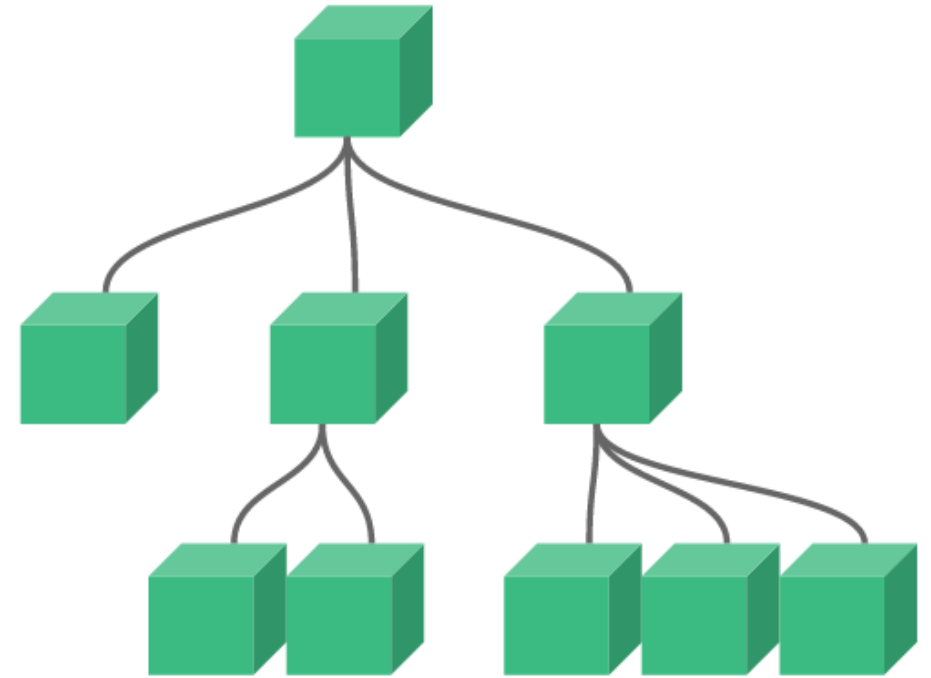
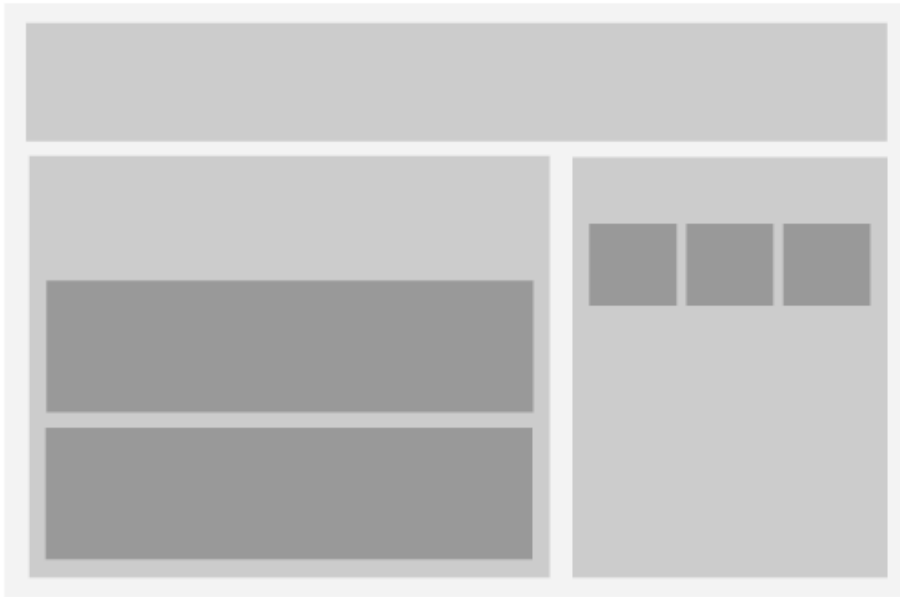
Læs og prøv til og med "Handling User Input"

ELLER

Følg youtube'en i moodle

[Vue JS 3 for beginners](#)





# Vue component basics

# Vue component basics

- Nedbryder koden i små overkommelige bidder
- Åbner døren for kodegenbrug
- Kan være alt fra en simpel knap til en hel side
- Components kan bruge andre components
- Ingen direkte forbindelse eller delt tilstand mellem components

1. **Vue component basics**
2. 1-2-3 Component
3. Eksempel
4. Opgave



# 1-2-3 Component

1. Skriv HTML for et component
2. Registrer et component via `vue.component('mit-komponent-navn', { ... : ... })`
3. Brug dit component:  
`<mit-komponent-navn></mit-komponent-navn>`

## Vue component basics

1. Vue component basics
2. **1-2-3 Component**
3. Eksempel
4. Opgave



# Eksempel: Send data fra A til B

## Parent

```
<div id="app">
  <my-component v-bind:prop1="variabel1">
  </my-component>
</div>
```

```
new Vue({
  el : "#app",
  data : {
    variabel1 : "En værdi"
  }
});
```

## Child component

```
<div id="cmp" style="display:none;"><div>
  {{prop1}}
</div></div>
```

```
Vue.component('my-component', {
  template :
    document.getElementById("cmp"),
  props : ["prop1"]
});
```

# Opgave

## En vareliste - to components

Udover vue instansen laves to components:

Et item-component der viser en vare (title, beskrivelse og pris)

Et list-component som viser en overskrift og en liste af item-components

Indsæt et nyt item

# Dagens program

8.15: [Demo] Installation af Vue i visual studio

8.45: Repetition af Lektion 4

9.15: [Pause]

9.45: [Opgave]

10.25: [Oplæg] Events i Vue

11.15: [Pause]



## Vue event handling basics

# Vue event handling basics

- Husk – ingen direkte forbindelse mellem components
- Man kan sende data den ene vej via v-bind:prop – hvad med den anden vej?
- Her bruger man events

## Vue event handling basics

1. **Vue event handling basics**
2. 1-2-3 Event handled
3. Eksempel





# 1-2-3 Event handled

1. Definér event handler på parent component
2. Kæd event handler sammen med child component
3. Udløs event i child component

## Vue event handling basics

1. Vue event handling basics
- 2. 1-2-3 Event handled**
3. Eksempel



# Eksempel: Send data fra B til A

## Parent

```
<div id="app">
  <my-component v-on:dead="mourn">
  </my-component>
</div>
```

```
new Vue({
  el : "#app",
  methods : {
    mourn( data ) {
      console.log(data.msg);
    }
  }
});
```

## Child

### component

```
<div id="cmp" style="display:none;">
<button @click="die">Imma die</button>
</div>
```

```
Vue.component( 'my-component', {
  template :
document.getElementById( "cmp" ),
  methods :{
    die() {
      this.$emit("dead",{ msg: "so
sad" });
    }
  }
});
```

# Opgave

## Fjern en vare fra listen

Tilføj en "fjernknap" på jeres item-component, som sender et event til sin parent om, at varen gerne vil fjernes fra listen.

# Lektion 4 gjorde os klogere på Vue.js

Næste skridt hedder lektion 5: Debugging

# Debugging

Eller på godt dansk – aflusning!

# Debugging

- Aflusning mig her, aflusning mig der, aflusning alle vejene
- Debugger vi fortolket eller kompliceret kode?
- Debugger vi clientside eller serverside kode?

# Hvad tilbyder browseren?

- Start en browser navigér ind på <https://techcollege.dk>
- Tryk på F12 eller højreklik og vælg "inspect element"
- Orientér jer omkring udviklingsværktøjet

# Hovedpunkterne ved web-debugging

1. Hvad siger konsollen?
2. Sæt breakpoints i kildekoden
3. Tjek netværksdata
4. Hvilke CSS regler gælder?
5. Derudover kan browsere som regel simulere en touch-skærm med musen som finger.



# Vue Routing

/hvor/skal/vi/hen/du.html



Vejen til det rigtige indhold

# Baggrund

- Querystrings (skrammel.dk/?[product=lenovo-t590&page=2](#)) blev bla. brugt til produktsider og andet specifikt indhold.
- Men så sagde Google:



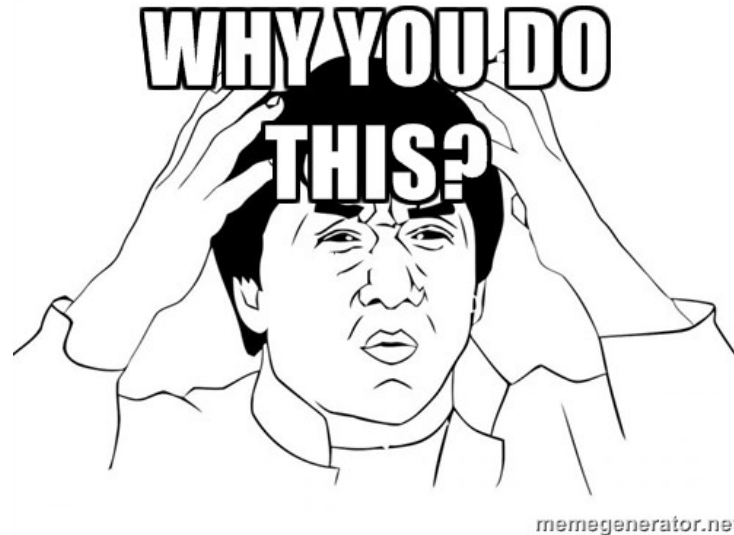
1. **Baggrund**
2. Routing 1 af 2
3. Routing 2 af 2
4. I en nøddeskal
5. Hvordan ser det ud i Vue?
6. #routingrules
7. recap



# Routing

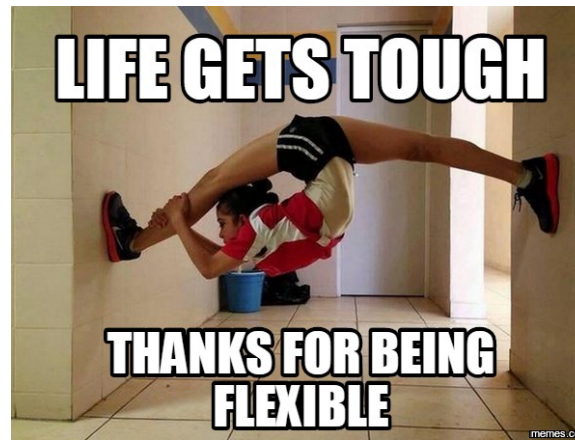
Slide 45

- Routing handler at lede trafik frem til en bestemt adresse
- I SPA (og web generelt) handler routing om at lede brugeren frem til bestemt web indhold fx en side.
  - Stien angives i URL'en fx <http://dr.dk/tv/programmer>
- Men **tv/programmer** er jo to mapper!
  - Nejnej, man lyver bare og leder i al hemmelighed trafikken frem til samme serverside kode, der så finder ud af, hvad du må kigge på.



memegenerator.net

# Fordi Jackie...



- Vi gør det for at
  - Have ét single point of entry i vores kode, og lade forretningslogik bestemme indholdets udseende.
  - Udnytte fx Model View Controller designmønstret
  - At være søgemaskineoptimerede (betale Google skat)
  - Navigation (frem og tilbage)
  - Ikke mindst være fleksible

# I en nøddeskal

Hvis – brugeren lander på denne sti:	Så – anvend et View for
/	En forside med udvalgte varer
/udsalg/	Liste over udsalgsvarer
/lenovo/t590.html	Produktdetaljer for pågældende produkt (model)
/om-os/	Vis side med tekst om virksomheden

1. Baggrund
2. Routing 1 af 2
3. Routing 2 af 2
- 4. I en nøddeskal**
5. Hvordan ser det ud i Vue?
6. #routingrules
7. recap



# Hvordan ser det ud i Vue.js?

Slide 48

- Components:  
(View)
- Stier:
- Nuværende sti:
- Controller:
- Tegnefunktion

```
JS
const NotFound = { template: '<p>Page not found</p>' }
const Home = { template: '<p>home page</p>' }
const About = { template: '<p>about page</p>' }

const routes = {
  '/': Home,
  '/about': About
}

new Vue({
  el: '#app',
  data: {
    currentRoute: window.location.pathname
  },
  computed: {
    ViewComponent () {
      return routes[this.currentRoute] || NotFound
    }
  },
  render (h) { return h(this.ViewComponent) }
})
```

# #routingrules

Slide 49

- Hvad kræver det?
  - En webserver konfigureret til at omdirigere trafik til én fil
  - **Eller:** brug #hashtag – se kilden <http://hoxer.net/routing.html> for eksempel på implementering i vue.js

- ASP.NET Core har det indbygget:

```
app.UseMvc(routes =>
{
    routes.MapRoute("default", "{controller=Home}/{action=index}/{id?}");
})
```

- Laravel (PHP) tilbyder det
  - `Route::get('/some/path/', function() {  
 return "Oh hai there!";  
})`



# Recap

Slide 50

- Routing forbinder en adresse med visning af webindhold
- Vi bruger routing til:
  - SEO (Google har aflivet Querystring)
  - Model-view-controller
  - Pæne og strukturerede URL'er
- Kræver at webserveren leder trafik til samme entry point
- Vi har set eksempler på routing