Web Frameworks Inleiding

Elektronica – ICT Sven Mariën (sven.marien01@ap.be)



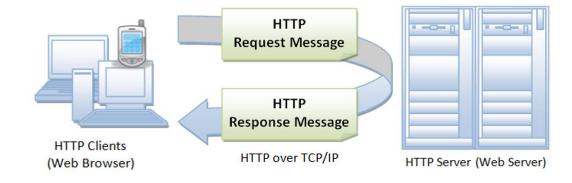
AP.BE

Web technologie

- Hoe werkt web technologie juist?
- Welke partijen spelen hier mee ?
- Welke communicatie vindt er plaats?

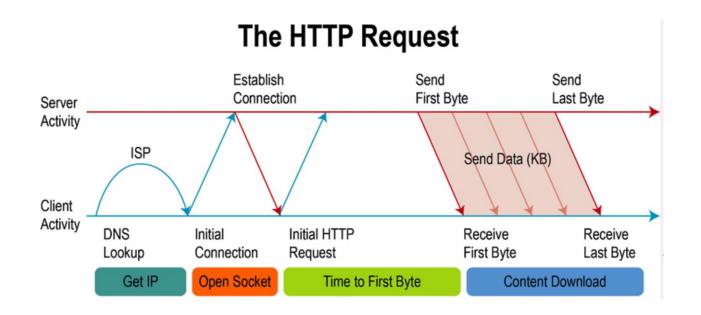


- Enkele sleutelwoorden:
 - Browser Web server
 - Request Response
 - HTTP HTTPS
 - TCP/IP port
 - URL
 - DNS



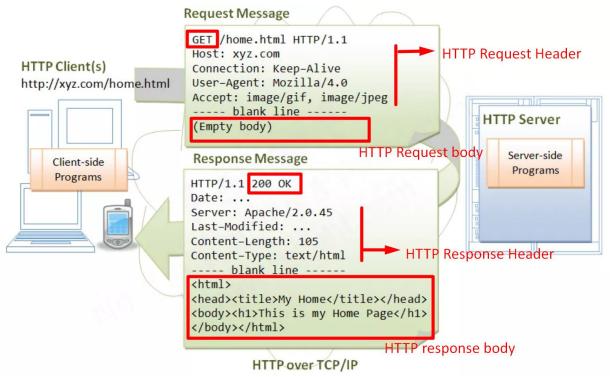


HTTP Request/Response





HTTP Request & Response in meer detail

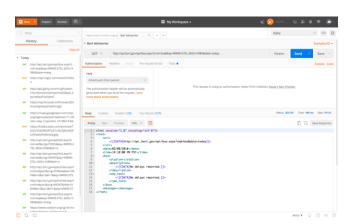


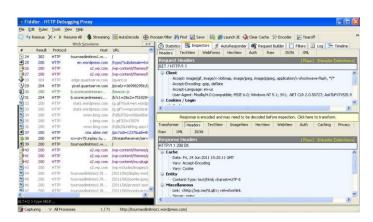
- HTTP Request definieert een aantal methods (verbs): GET, POST, PUT, DELETE,...
 - HTTP Response bevat een Status code (200, 404, 500,...) en content (HTML, CSS, javascript, font, XML data, JSON data...)



Tools

- Meekijken wat er juist wordt overgestuurd:
 - F12 (browser developer tools) !!
 - Buiten de browser:
 - Fiddler: http proxy (kan je dus mee ingrijpen)
 - Wireshark: op TCP niveau
- Zelf requests samenstellen en het resultaat analyseren
 - Postman







Web Servers

- https://nl.wikipedia.org/wiki/Webserver
- Luistert op een bepaalde poort (80,..) naar HTTP requests
- Levert de gevraagde resource terug aan de client.
- Welke content?
 - Statische content, zoals bestanden vanop disk (html, css, js, png, jpeg,...)
 - Dynamische content: gegevens die worden opgehaald uit een databank.
- Om gegevens op te halen hebben we een applicatie nodig aan de server zijde, dat kan bv. met PHP scripting, javascript,
 - => Dit noemen we server-side scripting



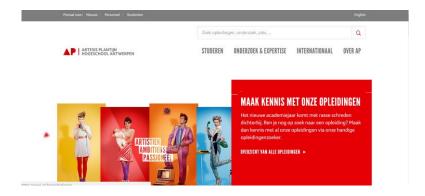




Weergeven van de HTML content

```
chtml lang-"nl" dir-"ltr" prefix-"content: http://purl.org/rss/1.0/modules/content/ dc: http://purl.org/dc/terms/ foaf: http://xmlms.com/firdfs: http://www.w3.org/2000/01/rdf-schema+ schema: http://schema.org/ sloc: http://rdfs.org/sloc/ins# sloct: http://rdfs.org/sloc/ins#
http://www.w3.org/2004/02/skos/core# xsd: http://www.w3.org/2001/XMLSchema# ">
     <(-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-36881676-1"></script>
          window.dataLayer = window.dataLayer || [];
          function gtag(){dataLayer.push(arguments);}
gtag('js', new Date());
          gtag('config', 'UA-36881676-1');
cscript>flowton(f,b,e,v,n,t,s)(if(f.fbq)return;n=f.fbq=function()(n.callMethod?
n.callMethod.apply(n,arguments):n.queue.push(arguments));if(!f__fbq)f._fbq=n;n.push=n;n.loaded=l0;n.version='2.0';n.queue=
cnoscripto.ting neight=1 wath=1 src= https://www.facedook.com/trid=2s05Jobsev430e4ang;ev=ageviewamp;moscript=2 for/noscripts
scripto.ffunction(w,d,t,u,n,a,m)(wf MauticfrackingObject*)=n;wfn]=wfn]|function()((w[n].q=wfn].ffl)).push(arguments),a=d.createElement(t)
[8]:a.async=i;a.src=u;m.parentMode.insertBefore(a,m))(window,document,'script','https://mautic.ap.be/index.php/mtc.js','mt'); mt('send', 'p
  cmeta name="title" content="Opleidingen | AP Hogeschool Antwerpen" />
 clink rel="canonical" href="https://www.ap.be/opleidingen" />
 cmeta name="description" content="AP Hogeschool Antwerpen heeft de opleiding voor jou. Je kan kiezen uit 24 professionele bachelors, 8 artis
opleidingen." />
 <meta name="Generator" content="Drupal 8 (https://www.drupal.org)" />
<meta name="MobileOptimized" content="width" />
cmeta name="HandheldFriendly" content="true" />
 cmeta name="viewport" content="width=device-width, initial-scale=1.0" />
 clink rel="shortcut icon" href="/themes/custom/ap hogeschool/favicon.ico" type="image/vnd.microsoft.icon" />
      <title>Opleidingen | AP Hogeschool Antwerpen</title>
```



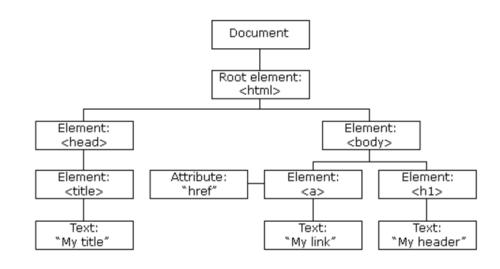






HTML -> DOM (Document Object Model)

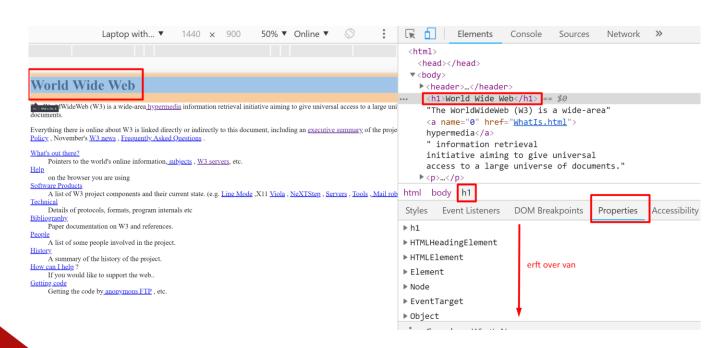
- De browser zet de HTML om naar een boomstructuur van 'objecten'
- Voor elk HTML element is er een overeenkomstige klasse
- Deze boomstructuur wordt vervolgens op het scherm weergegeven mbv. een 'Rendering Engine'
- Het DOM blijft 'in memory' zolang de betreffende pagina geladen blijft.
- Mbv. Javascript kan het DOM van de betreffende pagina worden aangepast.





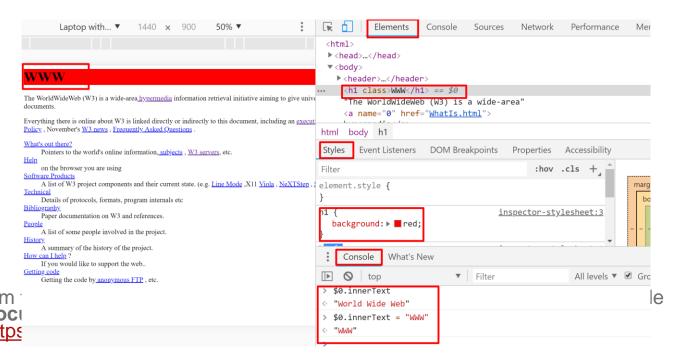
DOM bekijken

Via de "developer" tools (F12) kan je ook het DOM zichtbaar maken.





DOM aanpassen





Evolutie van het web (functionaliteit)

SPA

(Single Page Application): Enkel de initiele pagina wordt volledig ingeladen, de rest via ajax Native Mobile App: Een app specifiek ontwikkelt voor een bepaalt platform (iOs, Android,..) Progressive Web App (PWA): SPA die zich gedraagt zoals een app. Maar je moet deze (in tegenstelling tot de native en hybride apps) niet installeren via de Appstore / Market

Static Website:

Inhoud wijzigt niet, statische HTML Basic User Interaction: HTML Formulier / HTTP Post













Dynamic Website:

inhoud wordt op de server opgebouwd adhv. bv. Data uit een databank (producten, prijzen, afbeelding, beschrijving,...

-> server side script / code

Advanced user

interaction: mbv. Client side javascript kan er meer interactie zijn, bv. sneller feedback worden gegeven aan de gebruiker, .. Maar dit maakt het eveneens mogelijk om enkel stukken van een pagina op te vragen bij de server...



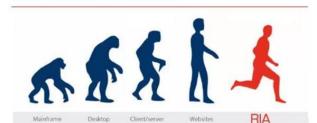
(Rich internet application): De interface wordt steeds uitgebreider, verschil met tradionele desktop apps wordt steeds kleiner

Hybrid Mobile App:

Gedraagt zich als een native app, maar is in weze een SPA/RIA die in een (onzichtbare) browser schil draait.



User Interface Evolution



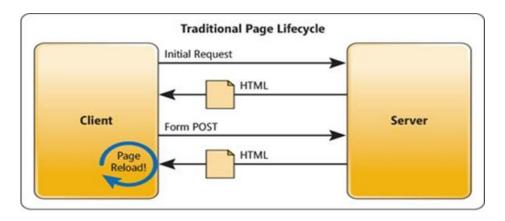


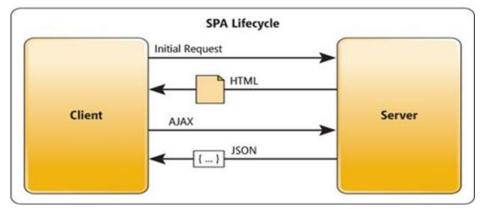
Eenvoudige SPA

• Om het principe van een SPA wat trachten te verduidelijken een heel eenvoudig voorbeeld:



Client / Server communicatie







AJAX

- · Afkorting voor: "Asynchronous Javascript and XML"
 - Asynchrone HTTP request naar de server, blokkeert bijgevolg de UI niet
 - De huidige HTML pagina blijft in de browser ingeladen in het DOM (wordt dus niet volledig ververst)
 - De AJAX call kan een stukje HTML terugkrijgen (bv. Een tabel,..) of enkel data in de vorm van XML,
 JSON,... (de benaming is dus enigszins misleidend)
 - Enkel het resultaat van de AJAX call wordt (via javascript) geïnjecteerd in het DOM van de huidige pagina.
 - Enkele voorbeelden van gebruik:
 - Ophalen van een andere "view" na een "button click".
 - Ophalen van gegevens terwijl de gebruiker gegevens invoert, bv. Lijst van landen,...
 - Validatie van de gegevens terwijl de gebruiker invoert, bv. Geldig bankrekeningnummer,...
 - Ophalen van meer data, bv. bij scrollen in een tabel met veel rijen,
 - •



Verschuiving in de architectuur van web toepassingen (Single Page

Static Website:

Inhoud wijzigt niet. statische HTML

Basic User Interaction: HTML Formulier / HTTP Post

Application):

Enkel de initiële pagina wordt volledig ingeladen, de rest via aiax in de achtergrond

Native Mobile

App: Een app specifiek ontwikkelt voor een bepaalt platform (iOs. Android,...)

















Dynamic Website:

inhoud wordt op de server opgebouwd adhv. bv. Data uit een databank (producten, prijzen, afbeelding, beschrijving,...

Advanced user interaction:

mby. Client side iavascript kan er betere interactie ziin, by, sneller feedback worden gegeven aan de gebruiker...

RIA

(Rich internet application): De interface wordt steeds uitgebreider. verschil met tradionele desktop apps wordt steeds kleiner

Hybrid Mobile App:

Gedraagt zich als een native app. maar is in weze een SPA/RIA die in een schil draait.

Thin Client / Thick server

Thick Client / Thin server



Graduele verschuiving naar "Thick Client"

- Verschillende behoeftes ontstaan hierbij:
 - Meer (javascript) code aan de client zijde => goede onderhoudbare structuur nodig
 - Behoefte om te kunnen 'databinden' (zie WPF), de HTML wordt veelal op de client aangemaakt en moet worden gekoppeld met data vanuit de server.
 - Complexere client toepassingen die door meerdere ontwikkelaars moet kunnen worden onderhouden vraagt om opdeling in kleinere blokken ('componenten')
 - 'Caching' van gegevens aan de client zijde en delen van gegevens tussen die componenten.
 - (Automatisch) testen van die client-side code (unit testing)
 - ...

=> Ontstaan van zogenaamde "Client Side Web frameworks"



Wat is een Web Framework?

- "A web framework (WF) or web application framework (WAF) is a software framework that is designed to support the development of web applications including web services, web resources,... "
- "Web frameworks provide a standard way to build and deploy web applications."
- "Web frameworks aim to automate the overhead associated with common activities performed in web development. For example, many web frameworks provide libraries for database access, templating frameworks, and session management, and they often promote code reuse."
- "Although they often target development of dynamic web sites, they are also applicable to static websites."
- Er bestaan:
 - server-side frameworks (thick server)
 - client-side frameworks (thick client)
 - 'isomorphic' frameworks



Client side frameworks (thick client): vooren nadelen

- Voordelen:
 - Minder server calls, lagere server load
 - lagere netwerk trafiek
 - lokaal cachen van data
 - Geen overhead / scherm refresh meer bij switchen tussen pagina's
 - Hogere gebruiksvriendelijkheid.
- Nadelen:
 - Initieel zal het mogelijk langer duren om de website te raadplegen (meer javascript en dependencies dat moet worden gedownload)
 - De site is niet meer zo toegankelijk voor Web crawlers (gebruikt door zoekmachines)
 - Bepaalde code moet dubbel worden geschreven (bv. Bepaalde client-side validatie in javascript en dezelfde validatie server-side in .NET)



Web frameworks: enkele voorbeelden

- Server side:
 - ASP/MVC (Microsoft)
 - Express / Node.js
 - Ruby on Rails
 - Django
- · Client side:
 - Angular (Google)
 - ReactJS (Facebook)
 - Meteor
 - Ember
- Isomorphic:
 - Angular Universal
- App development
 - Hybrid apps: Cordova/Ionic (Based on Angular) of Angular Mobile Toolkit
 - Native apps: React Native, Nativescript (ook in combinatie met Angular),...
- Desktop applications
 - Electron (in combinatie met Angular, volledige hergebruik van codebase mogelijk)





React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES











Angular client side framework (Google)

- Momenteel: versie 12 (Mei 2021) (https://angular.io/)
- Opvolger van versie 2 (er bestaat geen versie 3). Versie 4 tem. 11 zijn hiermee "backwards compatible".
- Versie 2 was een complete "re-write" (dus niet "backwards compatible" met versie 1)
- Let op: versie 1 is bijgevolg bekend onder de naam Angular.JS! (https://angularjs.org/)

Angular Version History

