

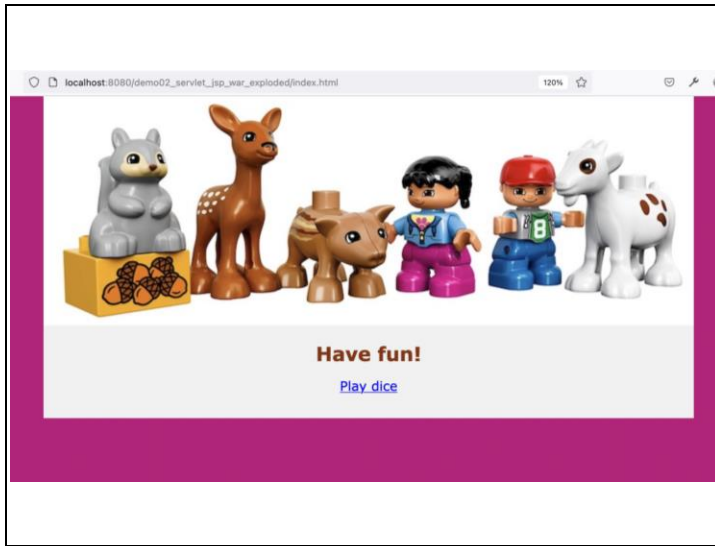
Web 2

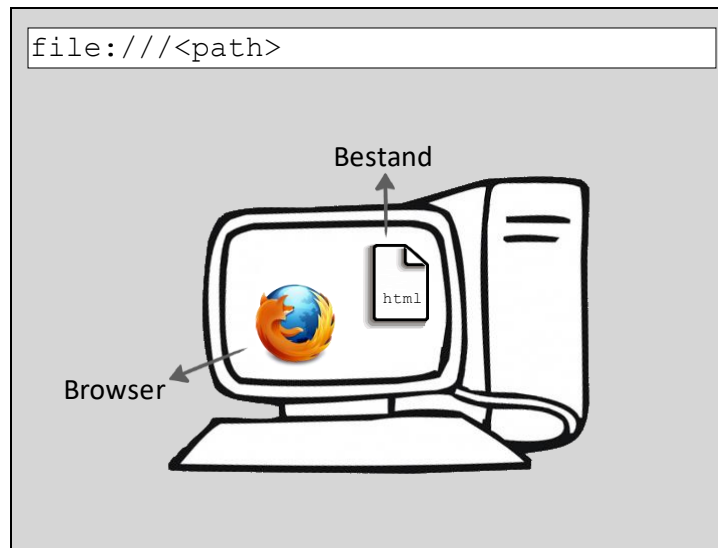
Les 1: Client Server

AGENDA

- ❑ Client-Servermodel
- ❑ HTTP - GET
- ❑ Foutopsporing
- ❑ Build en deploy

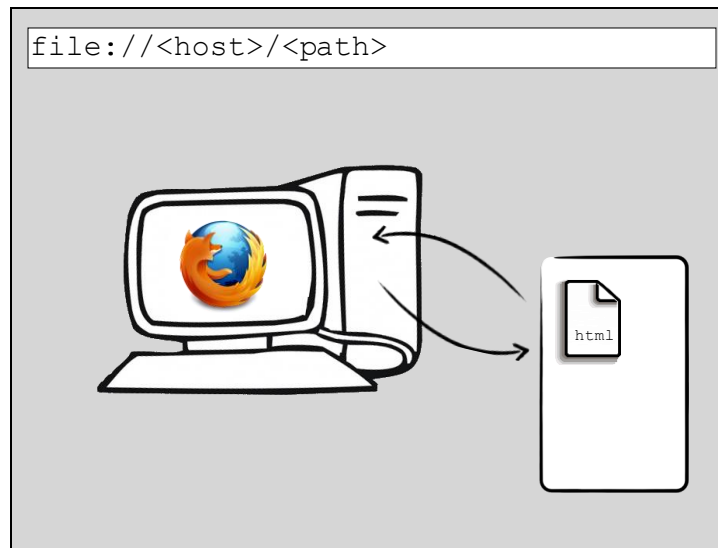
Inhoud: van statisch naar dynamisch





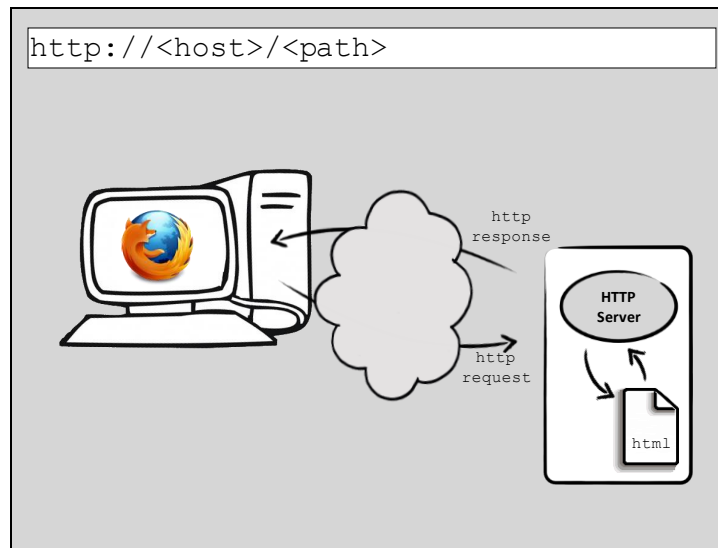
Via file-protocol:

- Browser
- Bestand op eigen pc



Via file-protocol:

- Browser
- Bestand op vaste plaats op andere PC in zelfde netwerk



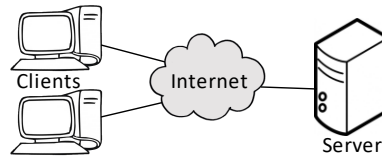
Via http-protocol:

- Browser
- HTTP Server

Bestand op vaste plaats en zelfde netwerk meestal moeilijk

—> http-protocol: bestand toegankelijk via internet

CLIENT SERVER MODEL



Het client server model beschrijft de relatie tussen twee computerprogramma's in een proces. Deze relatie is asymmetrisch. De cliënt verstuurt een service verzoek aan een ander programma, de server. De server verwerkt het verzoek. Hoewel het client-server principe kan worden gebruikt door programma's binnen een enkele computer, is het belangrijker binnen een netwerk.*

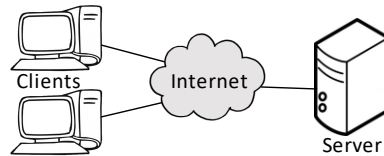
Vrij naar: <http://www.marqit.nl/wat-is-client-server>

HTTP werkt volgens het client-server model

Voorbeeld client: Browser (Chrome, Firefox, Internet Explorer, ...)

Voorbeeld server: Apache

CLIENT SERVER MODEL



heeft een service nodig	biedt een service aan
draait client programma	draait server programma
deelt geen resources	deelt resources met client
begint communicatie	wacht op vraag
vraagt (request)	antwoordt (response)

HTTP werkt volgens het client-server model

Voorbeeld client: Browser (Chrome, Firefox, Internet Explorer, ...)

Voorbeeld server: Apache

AGENDA

- ✓ Client-Servermodel
- HTTP - GET
- Foutopsporing
- Build en deploy

Vragen?



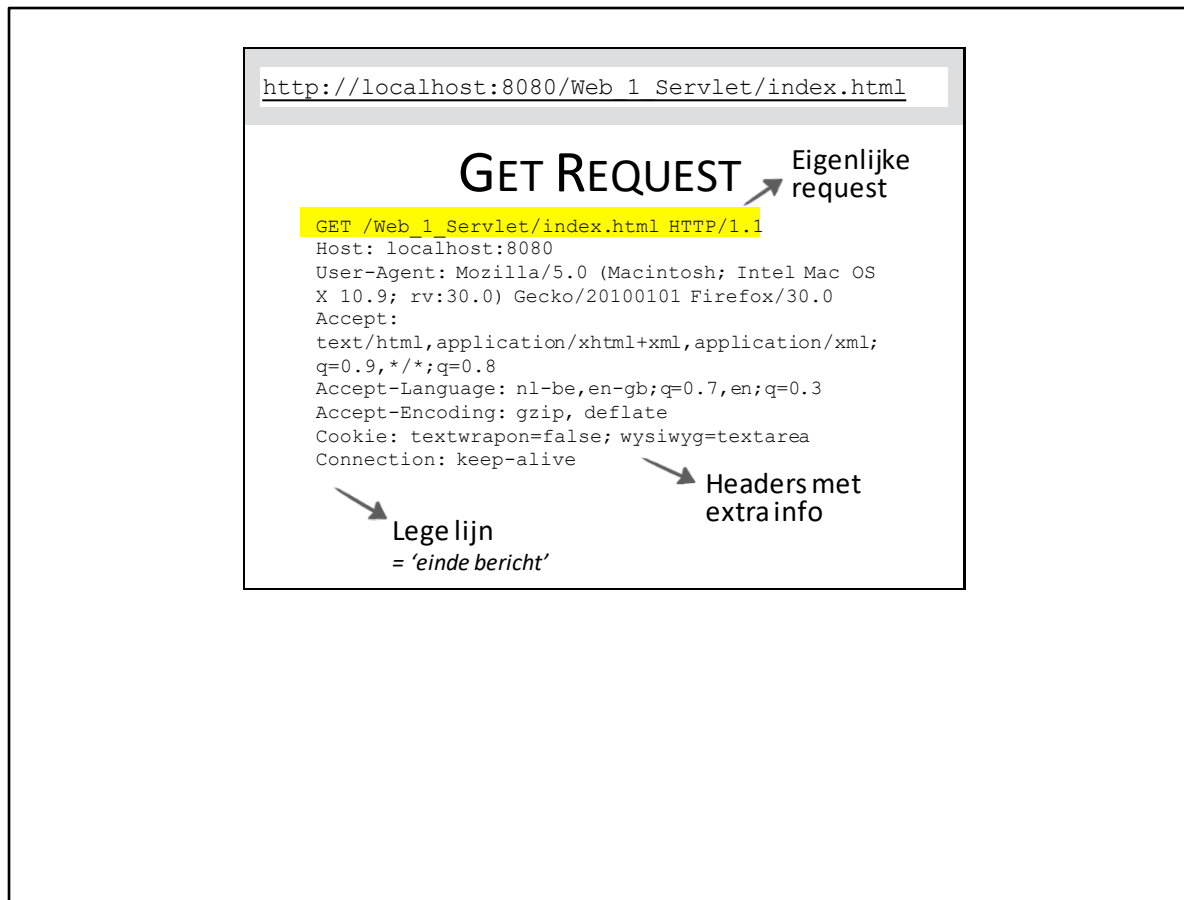
HTTP: protocol dat communicatie over het WWW beschrijft

- hoe boodschappen gevormd en overgebracht moeten worden
- hoe browsers en web servers hiermee moeten omgaan

Verschillende soorten boodschappen: Get, Post, Put, Delete, ...

Get en Post zijn de belangrijkste voor dit vak.

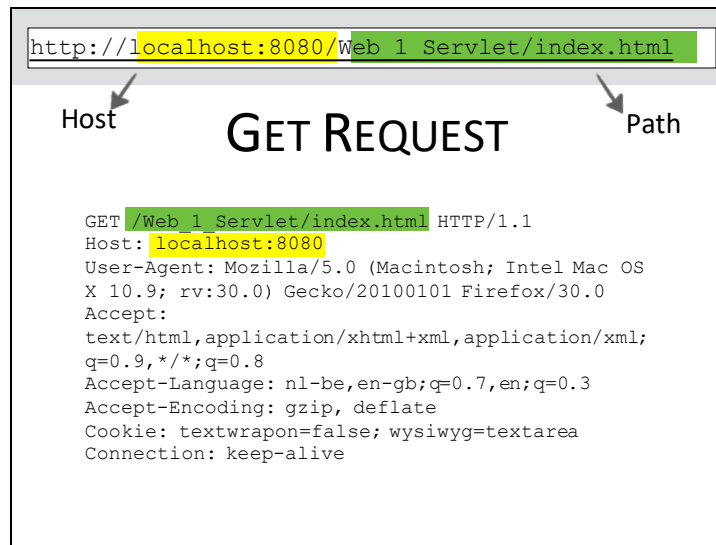
In de vakken computernetwerken 1 en 2 zal hier dieper op ingegaan worden



Wanneer je op een link klikt (of deze zelf intikt in de adresbalk van je browser) wordt een HTTP GET request verstuurd

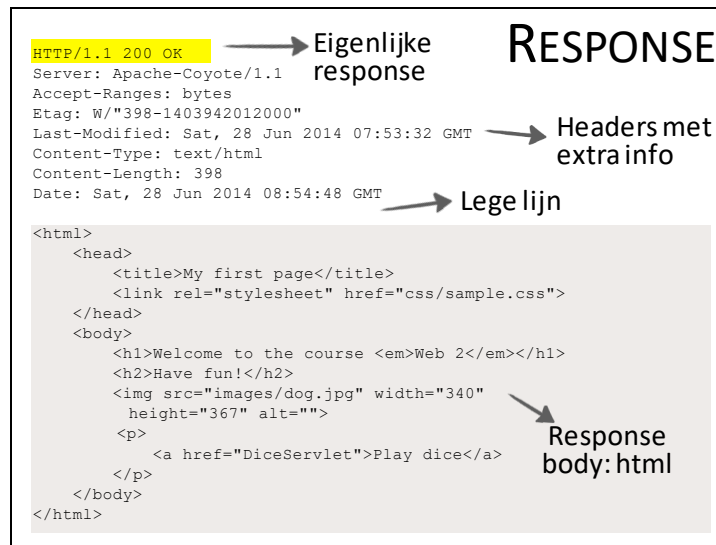
- zet je muis op een website
- klik rechts, kies "Inspect Element"
- Neem het tabblad "Network"
- Herlaad de pagina

- Klik op een request. Rechts verschijnen de details van die request.



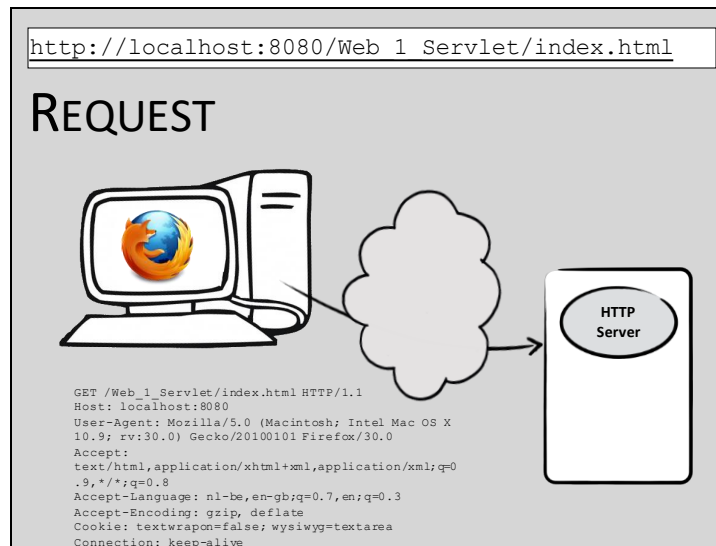
De link bestaat uit drie delen:

- Protocol (hier: http)
- Host (hier localhost)
- Path



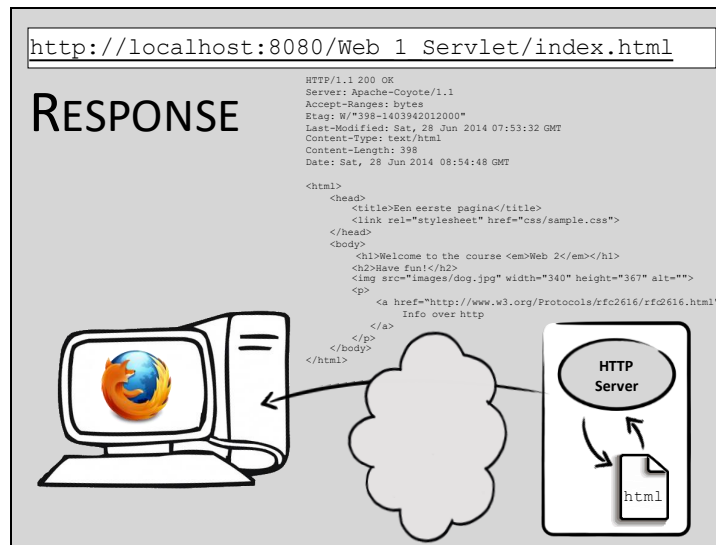
HTTP RESPONSE bevat

- Eigenlijke response
- Headers met extra info (oa meta informatie over de server)
- Lege lijn
- Response body (hier: HTML). De browser zal de HTML interpreteren en tonen op het scherm.



Samengevat...

Browser (client) stuurt HTTP GET request naar Web Server (HTTP Server)



De Webserver verwerkt de request en stuurt een response terug.
De client interpreteert de HTML die in de body van de response staat en toont die op het scherm.

GET

- Doel: data ophalen
- Afspraak: GEEN side-effect
- Link: `http://<host>/<path>`
- Inhoud bericht:

GET Request	Response
request lijn met URL	response lijn met status code
headers ^[1] _{SEP} met extra info	headers ^[1] _{SEP} met extra info
	body met bv. HTML

Een GET request wordt gebruikt om data op te halen. Na de verwerking van een GET Request zal de situatie op de webserver dezelfde zijn als voorheen. Er worden bijvoorbeeld geen data toegevoegd of gewijzigd.

AGENDA

- ✓ Client-Server model
- ✓ HTTP - GET
- Foutopsporing
- Build en deploy

Vragen?

FOUTOPSPORING



Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning.

Rich Cook

HTTP ERROR CODES

Code	Betekenis
1xx	Informational
2xx	Success
3xx	Redirection
4xx	Client Error
5xx	Server Error

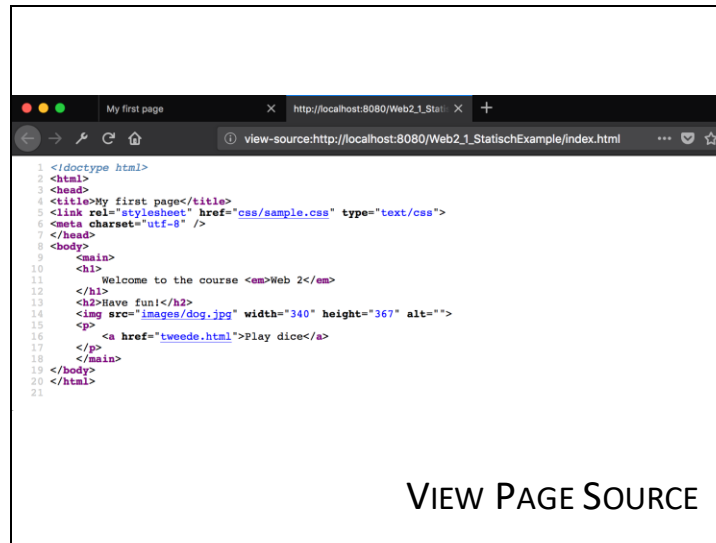
1xx - request is ontvangen, het proces gaat door

2xx - de request is ontvangen, begrepen, geaccepteerd en succesvol verwerkt

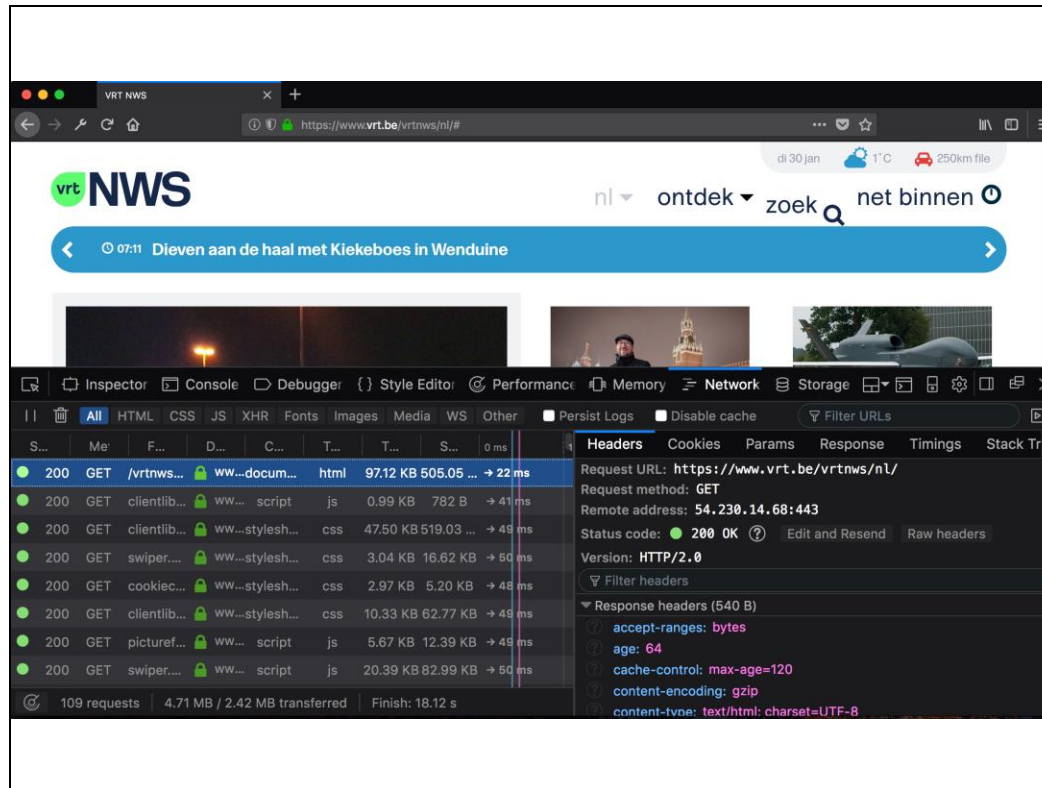
3xx - de client moet bijkomende actie ondernemen om de request af te werken

4xx - een foutboodschap die door de client veroorzaakt is, zoals het verkeerd typen van een URL

5xx - er is een fout opgetreden aan de kant van de server



Bij fouten: bekijk eerst de pagina-bron om foute HTML uit te sluiten.
Klik rechts op de webpagina en kies "View Page Source".



Bekijk de headers die meegestuurd worden met de http request:
Tools > Web developer > Network. Je ziet alle netwerkverkeer (kan ook in chrome, safari etc.)
Controleer het gevraagde pad, naam en waarde van parameters die meegestuurd worden, ...

INSTRUCTIES

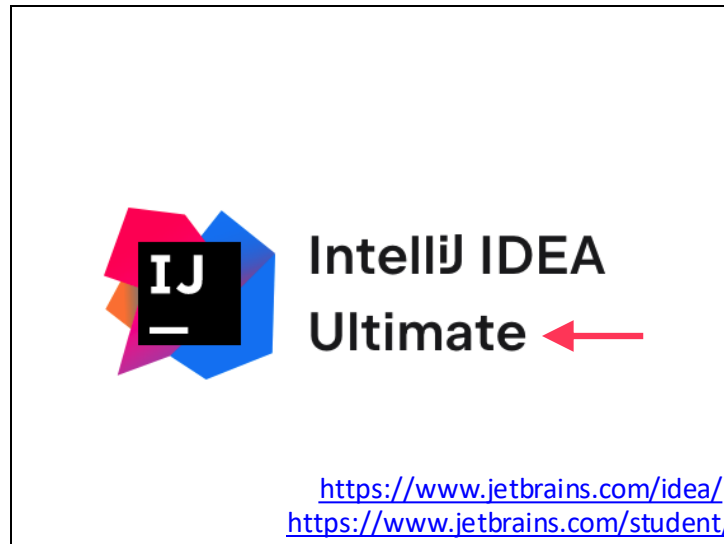
- ✓ Analyseer de **HTTP error code**
- ✓ Bekijk de code in **View Page source**
- ✓ Bekijk de request/response in **Developer Tools**

Belangrijk!!!

AGENDA

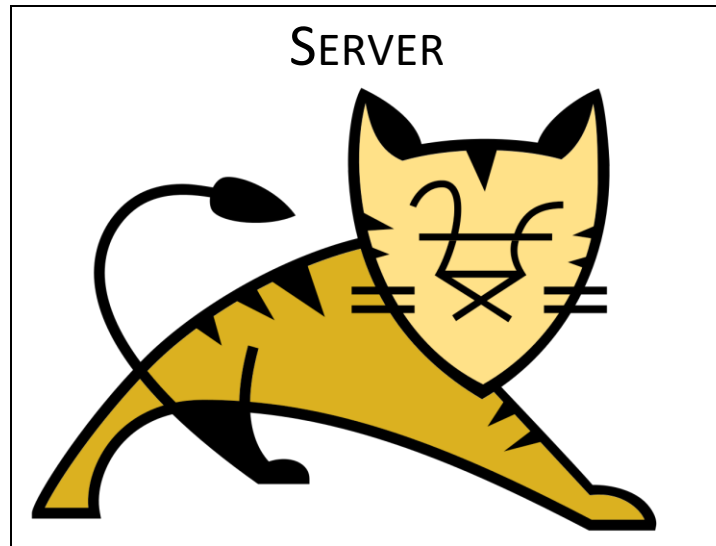
- ✓ Client-Server model
- ✓ HTTP - GET
- ✓ Foutopsporing
- Build en deploy

Vragen?



In de web-vakken gebruiken we IntelliJ Ultimate als IDE omwille van zijn populariteit in de industrie.

Vraag een student-account aan.

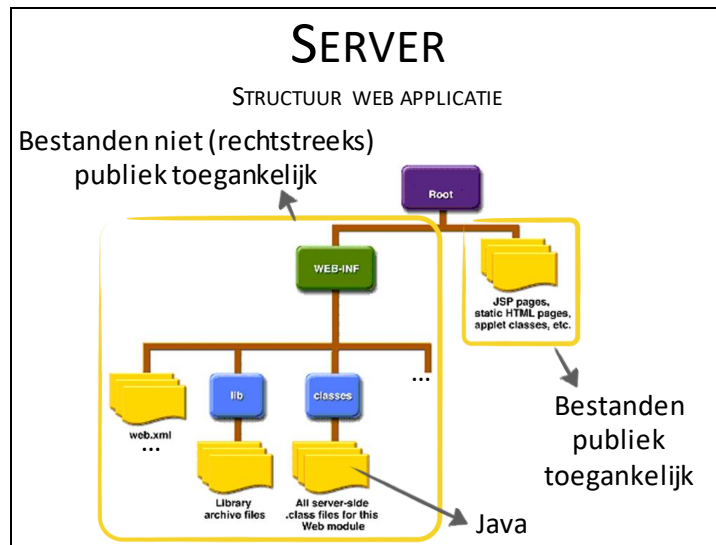


Een dynamische webapplicatie kan niet meer zomaar getoond worden in een browser. De Java-code moet immers ook verwerkt worden. Daarom gebruiken we een Tomcat-container. Deze software moet je downloaden en bewaren op je laptop (niet runnen).

<http://tomcat.apache.org/>

Gebruik Tomcat 9, niet 10.

In IntelliJ kan je je website lokaal runnen als je Tomcat gedownload hebt. School voorziet een externe webserver waarop Tomcat geïnstalleerd is zodat we onze projecten daarop kunnen runnen. Zie labo voor meer informatie.



Om een applicatie te deployen op een Tomcat server, moeten de bestanden een vooraf afgesproken structuur hebben. IntelliJ maakt die structuur voor jou.

VAN IDE NAAR SERVER

IDE (IntelliJ, ...)



1. Build

war file:

- web archive
- cfr.zip



2. deploy

webcontainer (Tomcat, ...)



Om van code naar gedeployede website te gaan, zijn er enkele stappen nodig.

Build

- java compileren
- bestanden in juiste folderstructuur zetten
- alles comprimeren —> War

Deploy

- war in de juiste folder van de web container zetten (Tomcat: webapps)

- unzip

- Tomcat manager: browse | deploy

—> zie labo

AGENDA

- ✓ Client-Server model
- ✓ HTTP - GET
- ✓ Foutopsporing
- ✓ Build en deploy

Vragen?