

Web Integration – 2TI

HC2 – Sessions & Cookies

HTTP

Statusloos

Connectieloos

Wat missen we?

Statusloos

De statusloosheid van de webserver betekent dat de server commando per commando afhandelt, zonder rekening te houden met voorgaande commando's. Hiermee wordt bedoeld dat aanvragen op de server geen invloed zullen hebben op toekomstige aanvragen.

Connectieloos

Dat betekent dat de HTTP server (de webserver) geen verbindingen zal onthouden of openhouden. Wanneer een client een pagina wil ophalen, dan verstuurt die een request. De server haalt het bestand op en verstuurt het met een response. Nadat de bestanden werden doorgestuurd, sluit de server de verbinding.

Wat missen we?

- Welke request komt van welke gebruiker?
- Welke gebruiker stuurt welke gegevens naar de server?
- Welke gebruiker mag gegevens opvragen?
- Hoe linken we bepaalde data aan een pagina?

Doel: We wensen de gegevens van bepaalde gebruikers te linken over verschillende requests heen.

Sessions & Cookies

- Cookies

- Laten toe om gegevens op de client te bewaren, en deze met een volgende request te versturen

- Sessions

- Maken het mogelijk om requests te linken aan gebruikers, en op de server gegevens per gebruiker te bewaren.
 - Steunt op session ID
 - Wordt opgeslagen in een cookie bij client
 - OF
 - Wordt doorgestuurd via POST of GET

Cookies

Definitie

Soorten

Gebruik

Cookies

- Laten toe om gegevens op de client te bewaren en deze met een volgende request te versturen.
- Code op onze pagina dat een cookie bewaart.
 - De aanvraag wordt meegestuurd met de HTTP response.
- Eens cookie bestaat op de client, geeft de gebruiker dit elke keer mee met een HTTP request.
- De server kan de cookie uitlezen en behandelen.
- Is leesbare tekst. Terug te vinden in de browser.

Cookies

- Cookies zijn geen
 - Scripts
 - Programma's
 - Programmeercode
 - Virussen
 - Spyware
- Cookies kunnen geen informatie van de harde schijf lezen, wijzigen of verwijderen.
- Cookies kunnen wel bijdragen tot ongewenste effecten, omdat ze toelaten een gebruiker te volgen op het internet.

Soorten Cookies

- Tijdelijke cookies
 - Cookies die gedurende één webrowsersessie worden bewaard.
 - De cookie zal dus verwijderd worden wanneer de gebruiker de webbrowser sluit.
Code op onze pagina dat een cookie bewaart. De aanvraag wordt meegestuurd met de HTTP response.
- Semi-permanente cookies
 - Cookies die gedurende meerdere webrowsersessies worden bewaard.
 - De cookie heeft wel een bepaalde vervaldatum
 - De cookie wordt verwijderd na die vervaldatum

Cookies aanmaken

Aangezien de cookie meegestuurd wordt met de response header, dient de functie aangeroepen te worden alvorens er enige vorm van HTML geschreven wordt.

Witruimte voor de `<?php ?>` tags telt als HTML uitvoer.

Tijdelijke Cookies

- Aanmaken met ingebouwde functie
- 2 argumenten: key, value
- Cookie wordt toegevoegd aan de HTTP response

```
setcookie('taal', 'NL');
```

Semi-permanente Cookies

- Aanmaken met ingebouwde functie:

```
setcookie(key,value)
```

- 3 argumenten: key, value, vervaldatum
- Vervaldatum: type integer. Time() functie gebruiken voor aantal seconden

```
//Cookie 7 dagen bewaren  
setcookie('taal', 'NL', time()+60*60*24*7);
```

Cookies gebruiken

- Indien een cookie meegestuurd wordt met een request, zal de superglobale `$_COOKIE` automatisch opgevuld worden
- Is een associatieve array
- Gelijkaardig gedrag aan `$_GET` en `$_POST`
- PHP parser haalt alle gegevens uit de cookie en vult de superglobale op
- Indien je op een pagina setcookie gebruikt, dien je te wachten op een nieuwe request met de nieuwe cookie data

```
$language = $_COOKIE['taal'];
```

Cookies verwijderen

- Terug gebruik maken van setcookie functie
- 3 mogelijkheden
 - Lege string meegeven: setcookie('language', '');
 - Op FALSE zetten
 - Tijd in het verleden plaatsen

```
setcookie('language', '');  
setcookie('language', FALSE);  
setcookie('language', '', time()-1);
```

Sessions

Definitie

Gebruik

Wat is een session?

- Een file met gegevens van een gebruiker
- Locatie: op de server
- Linkt requests aan bepaalde gebruikers
- Identificatie door session ID
 - Opgeslagen in cookie
 - In POST & GET request

Session controleren

- Elke session heeft een uniek session ID
- Wordt meestal toegekend aan een gebruiker
- Gebeurt standaard bij het opstarten van een session
- Session ID: Gegenerateerde tekencombinatie, door de server
- Wordt met elke request van de gebruiker meegestuurd, dankzij de cookie

Opslagen van de data

- Er wordt een file aangemaakt op de server
- Data wordt opgeslagen in key/value pairs
- Een file per session ID
- Naamgeving file is afhankelijk van server settings
- Kan ook in database opgeslagen worden

Session flow

- Client zonder session vraagt pagina op
 - HTTP request naar de server
- De pagina zelf bevat stukje code dat een session opstart
 - Genereren van uniek ID
 - Generatie van de file op de server
 - Stuurt standaard setcookie mee met het ID
 - Voorziet plaats in de file voor de data
- Server stuurt response terug, met de cookie
- Client stuurt nieuwe request, ditmaal met session ID, dankzij de cookie.
- Data wordt uit de file uitgelezen. Kan gebruikt worden

Session aanmaken

- Commando `session_start()`;
- Voert alles uit van de vorige slide
- Wanneer de client al een sessie heeft, haalt dit commando alle gegevens op en stelt ze ter beschikking
- Session data wordt in een superglobale geplaatst: `$_SESSION`
- `$_SESSION` kan worden gewijzigd => rechtstreekse opslag in de file op de server

Session aanmaken

```
session_start();  
  
if (isset($_SESSION["taal"])){  
    $taal = $_SESSION["taal"];  
}else{  
    $_SESSION["taal"] = 'nl';  
}
```

Session verwijderen

- Commando `session_destroy()`;
- Dit delete alle waarden uit de file
- Superglobale `$_SESSION` blijft wel bestaan
- Cookie bevat nog steeds het session ID
- Meer code nodig om alles volledig op te kuisen
- <https://www.php.net/manual/en/function.session-destroy.php>