T 5.1 PHP Sessies

# Inleiding

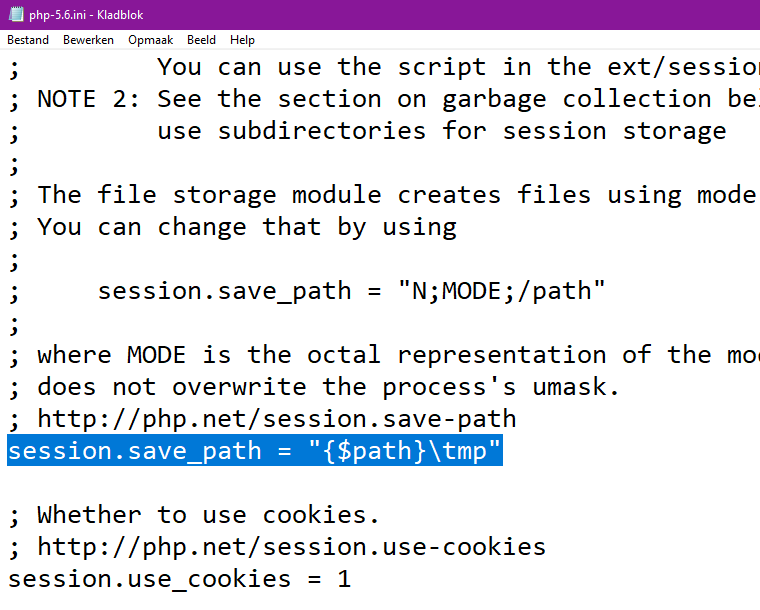
Met **PHP sessies** kan je gebruikersgegevens bewaren op de server, zodat je ze kan oproepen in alle scripts waarin ze nodig hebt.

Dat is handig, bijvoorbeeld bij een **winkelmandje** om te weten welke artikelen de bezoeker allemaal al toegevoegd heeft aan zijn mandje, ook na al zijn of haar omzwervingen door de verschillende pagina’s van je site, of zelfs op andere sites. Zolang de bezoeker zijn browser niet volledig afgesloten heeft, blijft zijn “historiek” (de toegevoegde artikels) beschikbaar.

Sessiegegevens worden **opgeslagen op de server**, in een bestand in een tijdelijke folder (meestal een “**tmp**” folder). De gegevens zijn dan beschikbaar voor alle scripts, dus voor alle pagina’s die de gebruiker bezoekt, en dat voor de duur van zijn bezoek, of tot wanneer de sessie (expliciet) beëindigd wordt.

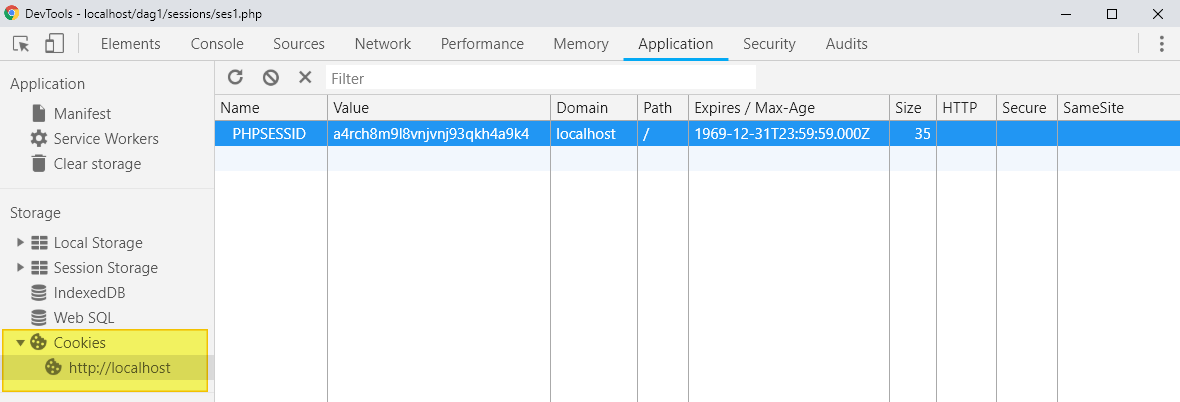
De plaats waar de gegevens worden opgeslagen – op de server dus – wordt bepaald door de instelling **session.save\_path** in **php.ini**, het configuratiebestand van PHP.

*php.ini*



Bij het starten van een sessie gebeurt het volgende:

* PHP genereert eerst een uniek **session\_id** voor de sessie, zoals “**a4rch8m9l8vnjvnj93qkh4a9k4**”. Dit kan je beschouwen als een willekeurig getal (groot genoeg om toevallige dubbels te vermijden).
* Dit **session\_id** wordt (meestal) opgeslagen in een **cookie** op de computer van de bezoeker, met als naam **PHPSESSID**.  
    
  In de **Inspector** van **Chrome** kan je dit cookie – en het **session\_id** – terugvinden, onder **Application -> Storage -> Cookies**.



* Op de server wordt er ook een bestand opgeslagen met als naam het **session\_id** voorafgegaan door **sess\_**, dus bv. **sess\_a4rch8m9l8vnjvnj93qkh4a9k4**. Dit bestand wordt dus opgeslagen in de tijdelijke folder.

Als een PHP script een bepaalde sessievariabele nodig heeft, bv. **$\_SESSION[“username”]**, gebruikt PHP het door de browser automatisch doorgestuurde **session\_id** (uit het **PHPSESSID** cookie) om het juiste bestand op de server te openen, en daaruit de nodige informatie te halen.

Een sessie kan **beëindigd** worden op diverse wijzen:

* als de bezoeker zijn browser (en alle vensters daarvan) volledig afsluit
* door een PHP script (met **session\_destroy()**, zie verder)
* na een timeout (veelal na 30 minuten zonder actie van de gebruiker)
* …

# Session functies

|  |  |
| --- | --- |
| **session\_start()** | Een **nieuwe sessie** starten **OF** de **bestaande sessie** hernemen.  Vooraleer je gebruik kan maken van de **$\_SESSION** superglobal, moet je deze functie oproepen, bij elke request.  Bij het oproepen van ***session\_start()***, wordt de array **$\_SESSION** meteen opgeladen. |
| **session\_id()** | Het **session\_id** opvragen (of instellen). |
| **session\_destroy()** | De sessie vernietigen.   Opgelet:   * je moet de sessie eerst initialiseren met ***session\_start()*** om ze te kunnen vernietigen. Door ***session\_start()*** uit te voeren, wordt **$\_SESSION** opgeladen (vanuit het bestand in de tmp folder) * ***session\_destroy()*** verwijdert wel de sessiegegevens (het bestand in de tmp folder), maar **maakt $\_SESSION niet leeg**.  In dezelfde request (zelfs na de oproep van ***session\_destroy()***) zijn deze gegevens dus nog beschikbaar. Pas bij de volgende request zie je het resultaat. * Als je wil dat de sessiegegevens onmiddellijk verdwijnen, moet je **$\_SESSION** zelf leegmaken. Dat kan met **unset($\_SESSION)**. * het ***session\_id*** wordt wel meteen leeggemaakt; MAAR als je ***session\_start()*** nadien nog eens gebruikt, wordt **hetzelfde session\_id** gerecupereerd; als je dat wil vermijden, moet je er zelf voor zorgen dat je een nieuw ***session\_id*** krijgt. |
| **session\_regenerate\_id()** | Een nieuw session\_id aanmaken. **De link met de vorige sessie is dan compleet verbroken.** |