

# *Web Expert 2*

---

Deel 2: Installatie



# PHP

---

Om PHP te kunnen gebruiken, dienen we eerst PHP lokaal te installeren.

Om PHP code uit te voeren hebben we in principe enkel een tekst editor nodig. In deze cursus gaan we in eerste instantie gebruik maken van Visual Studio Code. Maar we installeren ook een editor specifiek gebouwd voor PHP: PHPStorm



# *Installatie PHP*

Ga na of PHP reeds is geïnstalleerd op jouw toestel. Dit kan je doen door in een Command Prompt of Terminal venster het volgende commando uit te voeren: `php -v` of `php --version`.

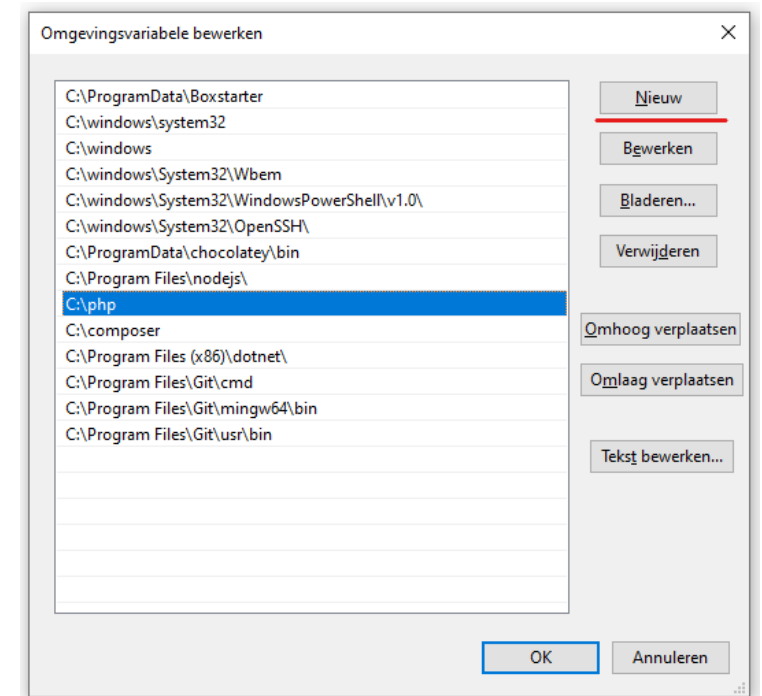
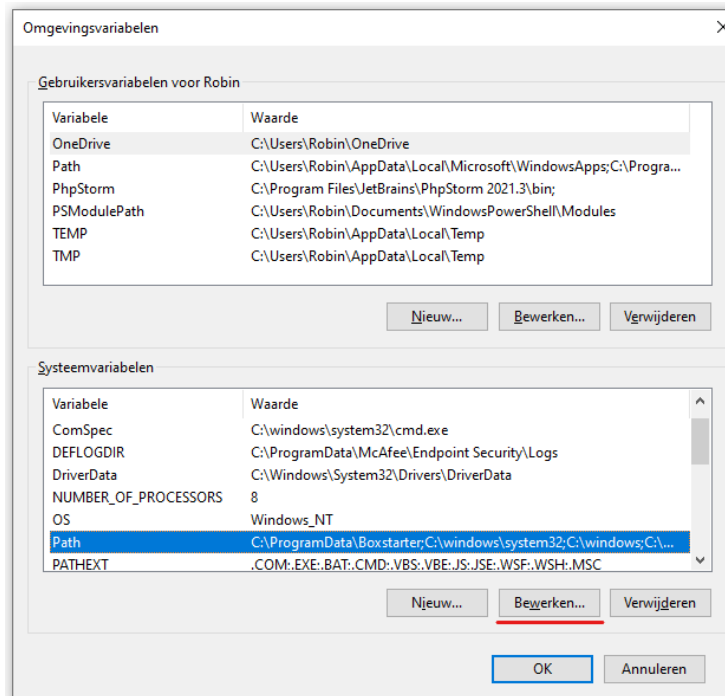
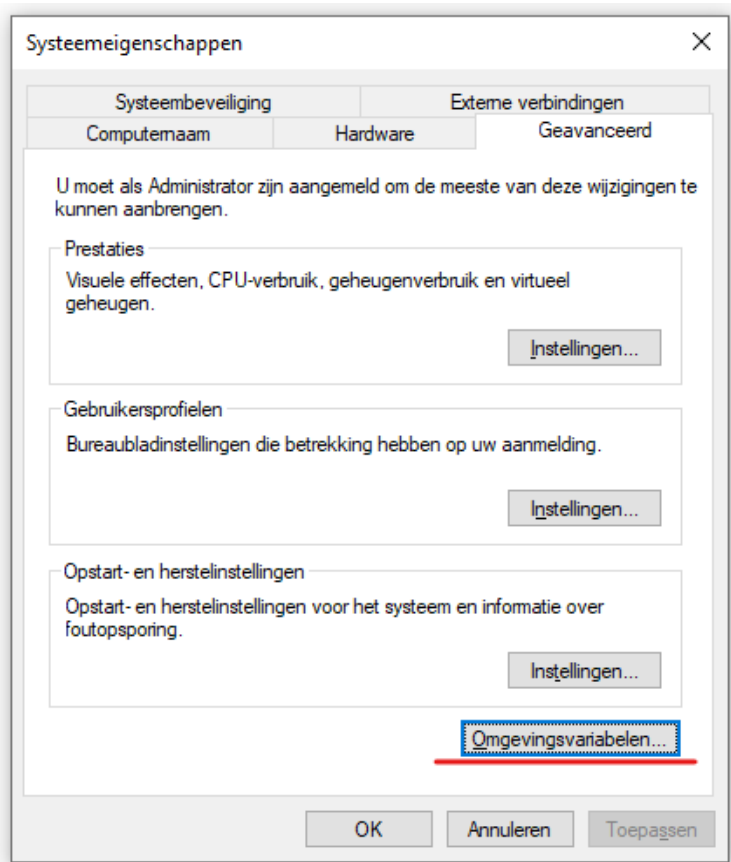
Zorg ervoor dat je minstens versie 7.4 geïnstalleerd hebt.

Het je geen of een oudere versie van PHP geïnstalleerd, installeer dan de laatste stabiele versie (momenteel 8.1). Opgelet de installatie voor Windows en Mac verschilt.

Daarom behandel ik eerst Windows en dan Mac.

# *Installatie op Windows*

1. Om PHP te installeren dienen we het ZIP pakket te downloaden.  
Voor Windows kan dit vanop: <https://windows.php.net/download#php-8.1>  
Opgelet, installeer de x64 Thread Safe versie.
2. Pak, eens gedownload, de bestanden uit op het volgende pad: C:\php
3. Om ervoor te zorgen dat Windows de PHP executable kan vinden, dienen we de PATH variabele aan te passen.  
Dit kunnen we doen door de volgende flow te volgen:  
Configuratiescherm > Systeem > Info > Geavanceerde systeeminstellingen > tabblad geavanceerd > Omgevingsvariabelen



# *Installatie op Mac*

1. We maken gebruik van de package manager Homebrew om PHP te installeren.  
Heb je nog geen Homebrew, dan kan je hier meer informatie terugvinden: <https://brew.sh/>.  
(Indien je iets met Homebrew installeert is het goed om eerst de volgende commando's uit te voeren:
  - `brew update`
  - `brew doctor`Hiermee zorg je ervoor dat Homebrew up to date is en je de laatste versie van pakketten binnen haalt.
2. PHP installeren kan dan met volgend commando: `brew install php`. Dit zal automatisch de laatste stabiele versie van PHP installeren.
3. Standaard staat er op bepaalde Mac toestellen al een PHP versie geïnstalleerd. Indien je een nieuwe (of oudere) versie wil installeren kan je dit bijvoorbeeld doen met `brew install php@7.4`.
4. Jouw Mac zal echter altijd de versie gebruiken die gedefinieerd staat in je shell script. Je kan deze versie nu op verschillende manier aanpassen.

Stel dat PHP 7.4 de versie is die momenteel gebruikt wordt.

Je kan dan het link commando van Homebrew gebruiken om de versie aan te passen:

- `brew unlink php@7.4`
- `brew link php@8.1`

Je zal dan gevraagd worden om een script te runnen:

```
echo `export PATH="/opt/homebrew/opt/php@8.1/bin:$PATH"` >> ~/.zshrc
```

Als je nadien opnieuw `php -v` uitvoert, zou de versie moeten aangepast zijn.

Is die niet geval dan dien je rechtstreeks je shell script aan te passen:

- `nano ~/.zshrc`
- Verwijder de oude lijn die verwijst naar de vorige instantie van PHP en voeg de volgende lijn toe:  
`export PATH="/opt/homebrew/opt/php@8.1/bin:$PATH"`

```

908 ;
909 ; Notes for Windows environme
910 ;
911 ; - Many DLL files are located in
912 ;   extension folders as well as the
913 ;   Be sure to appropriately set the e
914 ;
915 ;extension=bz2
916 extension=curl
917 ;extension=ffi
918 ;extension=ftp
919 extension=fileinfo
920 extension=gd
921 ;extension=gettext
922 ;extension=gmp
923 ;extension=intl
924 ;extension=imap
925 ;extension=ldap
926 extension=mbstring
927 ;extension=exif      ; Must be after mbstr
928 extension=mysqli
929 ;extension=oci8_12c  ; Use with Oracle D
930 ;extension=oci8_19   ; Use with Oracle D
931 ;extension=odbc
932 extension=openssl
933 ;extension=pdo_firebird
934 extension=pdo_mysql

```

# *php.ini*

---

Je kan PHP configureren op basis van het php.ini bestand. Hierin kan je de serve configuratie onder brengen, aangeven welke extensies moeten gebruikt worden, ....

Dit is een groot bestand en schrikt meestal af, maar is een bestand waar je sowieso gebruik van moet gaan maken. Bijvoorbeeld bij het optreden van errors, ....

Initieel zal dit bestand niet bestaan bij een Windows installatie. Hiervoor kan je de php.ini-development kopiëren en de naam van het gekopieerde bestand wijzigen in php.ini. Op zich staat de basisconfiguratie dan goed.



# *Installatie PHPStorm*

Als student kan je gratis gebruik maken van de IDE's en editors van JetBrains. Voor de ontwikkeling met PHP gaan we gebruik maken van PHPStorm.

<https://www.jetbrains.com/community/education/#students>

# Composer

In Web Expert 1 hebben we gebruik gemaakt van node.js. Een standaard node.js applicatie konden we uitbreiden met modules. Eén manier om deze modules toe te voegen was door gebruik te maken van npm pakketten.

Binnen PHP bestaat iets gelijkaardigs. Hier noemt dit Composer:

<https://getcomposer.org/>

Composer is dus een tool om dependencies te installeren en te managen binnen PHP.

# *Installatie*

Met `composer --version` kan nagaan of en welke versie van composer je hebt geïnstalleerd. Wij werken met composer 2 of hoger.

De installatie verschilt opnieuw van besturingssysteem en de documentatie hieromtrent kan je terug vinden op <https://getcomposer.org/download/>

Windows:

1. Download Composer-Setup.exe
2. Voer het net gedownloadte bestand uit.

Composer zal gedownload worden en er zal een pad variabele toegevoegd worden, zodat je het composer commando kan gebruiken.

Mac:

1. Open een Terminal in een map naar keuze
2. Voer stap 1 van het command line script uit. Dit zal de installer downloaden in de huidige map.
3. Voer stap 2 van het script uit. Dit zal een security check doen op de installer.
4. Voer stap 3 van het script uit. Dit zal de installer uitvoeren.
5. Voer stap 4 van het script uit. Dit zal de installer opnieuw verwijderen.

We gaan composer nu verplaatsen naar een map die in de PATH variabele gedefinieerd staat. Open daarom de composer map. Hierin staat een bestand `composer.phar`. Voer vervolgens volgend commando uit:

```
sudo mv composer.phar /usr/local/bin/composer
```

Composer zal nu overal aanspreekbaar zijn.

# Gebruik

Net zoals bij npm zitten de pakketten gebundeld op 1 plaats:

<https://packagist.org/>

In volgend voorbeeld tonen we het gebruik van Composer aan:

Maak een nieuwe map 'composer\_voorbeeld' aan en open deze map met PhpStorm.

Om Composer te gebruiken in het project is een `composer.json` bestand het enige wat je nodig hebt. In dit bestand worden alle dependencies van het project gedefinieerd. Standaard bevindt dit bestand zich in de root van het project.

In het bestand definiëren we onder de require key, welke dependencies we in het project willen. Als voorbeeld gaan we een logger installeren:  
monolog/monolog (<https://packagist.org/packages/monolog/monolog>)

```
{  
  "require" : {  
    "monolog/monolog": "2.*"  
  }  
}
```

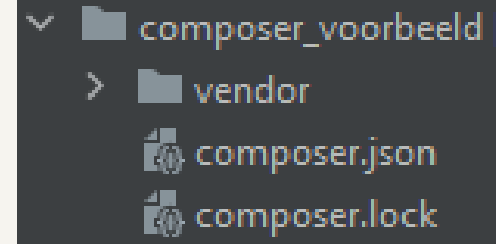
Aan de require key geven we vervolgens een object mee waarin de namen van de pakketten worden gemapt op de versie. Composer zal de pakketten dan opzoeken in de repository die onder de repositories key staat gedefinieerd. Wordt deze niet expliciet gedefinieerd, dan zal er gezocht worden op packagist.org.

Om deze dependencies te installeren gaan we het update commando gebruiken. In de root van je project voer je `composer update` uit. Dit commando zal nu een aantal dingen in orde brengen.

Er zal een `composer.lock` bestand aangemaakt worden waarin alle dependencies zullen geschreven worden volgens hun aangegeven versie. Het is dan ook van belang om het `composer.lock` bestand mee te committen zodat iedereen die met het project werkt, met dezelfde dependencies werkt.

Daarna zal het `install` commando uitvoeren. Hiermee zullen de dependencies in een `vendor` map geïnstalleerd worden.

De structuur van je project ziet er dan als volgt uit:



Wanneer er reeds een `composer.lock` bestand in het project aanwezig is, kan je gewoon het `composer install` commando runnen.

Eens je composer structuur opstaat, kan je nieuwe dependencies installeren met het `composer require` commando. Als voorbeeld gaan we een Chalk pakket installeren: `composer require tdtrung17693/php-chalk`

Ik vermeld ook even als sidenote dat wanneer je een volledig project in php van de basis aan opbouwt, de mapping van de klassen van belang is (op basis van de namespaces).

Composer voorziet hiervoor een autoload bestand. Hierin staat gedefinieerd wat er moet ingeladen worden. Een commando dat hierbij van belang is, is `composer dump-autoload`. Wanneer je een nieuwe klasse aanmaakt, en de mapping in orde wilt brengen, kan je dit commando uitvoeren. Meer info hierover kan je vinden op <https://getcomposer.org/doc/01-basic-usage.md#autoloading>