

## Lesdoelen

- ✓ Je kent de Model View Controller software architectuur en begrijpt waarvoor deze gebruikt wordt.
- ✓ Je begrijpt de taak/rol van de Controller binnen MVC en kunt een Controller bouwen voor een blog.
- ✓ Je begrijpt de taak/rol van de Model binnen MVC en kunt een Model bouwen voor een blog.
- ✓ Je weet wat een template engine is en waarvoor deze gebruikt wordt

# Model View Controller

## Inleiding MVC

Model View Controller (MVC) is een ontwerp patroon welke verschillende verantwoordelijkheden binnen je (php) applicatie opdeelt in drie separate onderdelen. Deze onderdelen zijn: data (model), presentatie (view) en logica (controller). Hieronder worden de verschillende eenheden uitgelegd:

### Model

Het model representeert je data structuren binnen de applicatie. Voor een MVC applicatie schrijf je vaak meerdere model classes, voor elk type data. Deze welke bijvoorbeeld functies bevatten om data op te halen of op te slaan in een database.

### View

De view representeert alles wat aan de eindgebruiker gepresenteerd wordt. Denk hierbij aan een HTML pagina met een overzicht van een winkelwagen of een RSS feed met de nieuwste producten. Vaak wordt binnen een MVC PHP applicatie een template engine gebruikt, de volgende les gaan we hier mee aan de slag. De view doet geen enkele verwerking zoals controles of berekeningen van de gegevens die getoond worden, dit moet in de model gebeuren.

### Controller

De controller dient als initieel aanspreekpunt en handelt de HTTP requests vanuit een browser af. De controller roept aan de hand van deze aanvraag het juiste model en view op om een web pagina te genereren en aan een gebruiker te tonen.

Door de scheiding van verantwoordelijkheden heeft je applicatie een duidelijke structuur en wordt de leesbaarheid en herbruikbaarheid van de code bevorderd. Er kunnen ook gemakkelijk meerdere mensen aan het zelfde project werken. Zo kan de een werken aan de verschillende model classes en de ander zich bezig houden met de HTML templates in de view.

# Model

## Voor dat we beginnen

Voor dat we het model gaan schrijven is het handig om een inventarisatie te maken van de functionaliteit die we nodig hebben.

## Functionaliteiten van een blog

Een blog heeft de volgende basis functionaliteit:

1. Een overzicht van de verschillende post's.
2. Een detail pagina van een entry.
3. Een pagina om een bestaande post aan te passen.
4. Een pagina om een nieuwe post aan te maken.

## Wat we nodig hebben in het model?

### Velden

Een post in een blog bevat de volgende velden:

- ✓ id
- ✓ titel
- ✓ body
- ✓ datum / tijd

### Acties

Je moet de volgende acties kunnen uitvoeren op de post in het blog:

- ✓ Overzicht
- ✓ Detail
- ✓ Verwijderen
- ✓ Bewaren
- ✓ Toevoegen

# Opdrachten

## Opdracht 1: xTemplate

De volgende les gaan we aan de slag met de template engine, xTemplate. Download xTemplate en lees alvast de documentatie door:

[http://sourceforge.net/project/showfiles.php?group\\_id=16714&package\\_id=14016](http://sourceforge.net/project/showfiles.php?group_id=16714&package_id=14016)

## Opdracht 2: Html maken

Voor de volgende les moet je HTML geproduceerd, zodat we die kunnen gebruiken in je blog.

Maak voor de volgende les de volgende html pagina's:

- ✓ Overzicht pagina met de titels en lees meer links van alle posts
- ✓ Detail pagina met de titel, body en datum

De html met volgens XHTML worden gemaakt (zonder frames) samen met vormgeving in css. In de volgende les gaan we deze pagina's ombouwen tot templates.