

# Yii level 1

- [01 Wat is Yii?](#)
- [02 Installation Yii](#)
- [03 Eerste CRUD](#)
- [04 Routing](#)
- [05 Menu](#)
- [06 Grid View](#)
- [07 En verder?](#)

# 01 Wat is Yii?

*In deze les gaan we leren wat een framework is en waarom we Yii gaan leren.*

*We leren verder wat MVC en CRUD is en wat heeft dat te maken heeft met een framework.*

Voor het examen moet je een webapplicatie bouwen.

Je moet zelf weten hoe je dat doet. De meeste studenten kiezen voor PHP. Waarom? Het is relatief eenvoudig en de meeste web applicaties zijn ontwikkeld met PHP ([bron \(Koppelingen naar een externe site.\)](#)).

Veel websites lijken op elkaar en doen vaak dezelfde dingen. Ze maken een verbinding naar de database, laten gegevens zien, je kunt aanloggen, je kunt soms gegevens veranderen en ga zo maar door.

Om niet telkens al deze 'standaard' zaken van 'scratch' af aan te moeten bouwen zijn er zogenaamde frameworks ontwikkeld. Elk framework is eigenlijk een verzameling bouwblokken die je telkens op nieuw kunt gebruiken voor jouw website. Alle frameworks zijn net een beetje anders, maar er zijn ook veel overeenkomsten.

Wat zijn belangrijke zaken als je voor een framework kiest:

1. Wordt het framework actief onderhouden; bestaat het al wat langer? Je wilt niet in iets gaan investeren dat volgend jaar misschien niet meer bestaat.
2. Hoe moeilijk is het om een framework te leren; wat is de complexiteit?
3. Hoe snel kun je iets ontwikkelen met het framework?
4. Hoe flexibel is een framework; kun je er echt alles mee maken?

Bij punt 3 en 4 is het vaak het een of het ander. Of je kunt heel snel iets bouwen en je bent niet zo flexibel, of je kunt er eigenlijk alles mee ontwikkelen maar het kost veel tijd om dat te doen.

Bijvoorbeeld: PHP is super flexibel je kunt er echt alle mee ontwikkelen, het Laravel framework is ook redelijk flexibel, het Yii framework is iets minder flexibel en als je naar bijvoorbeeld WordPress kijkt (wat je ook als een soort framework zou kunnen zien), dan is dat het minst flexibel. Aan de andere kant kun je met WordPress heel snel een site maken, je bent alleen een beetje beperkt in de functionaliteiten.

# Waarom Yii?

Laravel is het meest gebruikte PHP-framework. Yii staat in de meeste lijstjes ongeveer op plaats 7. Dat komt, omdat je weliswaar "snel even wat kan bouwen", maar je kunt het vrij lastig alles precies zo maken als jij het wilt. Het is minder flexibel en al je het precies zo wilt maken als jij precies wilt dan moet je meer moeite doen. Als je straks een project maakt dan kun je waarschijnlijk ongeveer 70%-80% van je code maken met Yii, daarvoor hoef je heel weinig te programmeren en heb je tijd om je te storten op de overige 20%-30%. De exprojecten amens die jullie gaan doen lenen zich goed om met Yii te maken. Je bent namelijk redelijk vrij in het ontwerp en kunt dus de 'Yii-manier' aanhouden.

Maar wat dan als ik later verder wil met (bijvoorbeeld) Laravel? Dat kan en Yii en Laravel hebben dezelfde opzet, MVC. Je zult in Laravel dus veel dingen terugvinden die je in Yii hebt geleerd.

## Laravel versus Yii

Voor een uitgebreide vergelijking tussen deze twee frameworks verwijs ik naar:

<https://keyua.org/blog/laravel-vs-yii-comparison/>

## MVC?

MVC, wat is dat? MVC staat voor Model, View en Control. Deze drie-deling wordt gemaakt in je code, zodat je eenvoudiger code kunt terugvinden.

|         |  |
|---------|--|
| Model   | Hierin staat waar de verbinding tussen jouw code en de database beschreven   |
| View    | Dit is de front-end en is vooral HTML/CSS (bootstrap) met hier een daar een 'vleugje' PHP  |
| Control | Dit is waar je bepaald wat er gebeurt, als je op een knop drukt ergens dan wordt de code uit de bijbehorende control uitgevoerd. |

In een plaatje ziet dat er als volgt uit. De gebruiker doet wat en stuurt daarmee een verzoek (request) naar de juiste controller. De controller 'overlegt' met het model als er iets met data moet

gebeuren en het resultaat wordt via de view aan de gebruiker getoond.

image-1594413597727.png

In de introductie van deze module (workshop) komen deze begrippen terug. Als het nu allemaal nog wat onduidelijk is, geen probleem we gaan er mee aan de slag en zullen snel een gevoel krijgen wat MVC precies is en hoe het werkt.

[Uitleg MVC YouTube \(EN\)](#)

image not found or type unknown



## CRUD?

Nog even....waar staat CRUD ook alweer voor? CRUD staat voor **C**reate, **R**ead, **U**ppdate en **D**elete. In een CRUD-applicatie kun je dus gegevens aanmaken (create), lezen (read), bewerken (update) en verwijderen (delete).

Voor jouw examen moet je minimaal een CRUD kunnen maken.

Yii heeft een CRUD-generator en je kan zonder te coderen een standaard CRUD-applicatie maken. Super handig dus voor het examen. In de volgende les gaan we hier gelijk mee aan de slag.

## Quiz

We gaan even een korte kennis quiz doen en gaan je bevragen over MVC, CRUD, en Frameworks. Als je de tekst op deze pagina goed hebt gelezen dan kun je de kennis quiz maken.

# 02 Installation Yii

*In de komende paar opdrachten gaan we alles klaarmaken en installeren voor het werken met Yii. Aan het eind van de opdracht 4 heb je volledig werkende Yii CRUD-applicatie.*

Een CRUD-applicatie is een applicatie waarmee je records kunt aanmaken (Create), records kunt lezen (Read), records kunt aanpassen (Update) en records kunt verwijderen (Delete).

We gaan drie stappen doen om een werkende web app te maken:

1. installeren van alle software
2. maken van een eenvoudige database
3. generen van code

In drie stappen hebben we straks een *werkende* web app in PHP gemaakt.

Voordat je met deze les begint weet je hoe je met [phpMyAdmin \(Koppelingen naar een externe site.\)](#) (of ander tool) een database kunt maken en hoe je een .sql file kan importeren. Ook heb je XAMPP geïnstalleerd.

## Installatie

### XAMPP

Je hebt [XAMPP \(Koppelingen naar een externe site.\)](#) geïnstalleerd. Voor Yii hebben dit nodig, omdat we een database gaan gebruiken.

Heb je nog geen XAMPP of werkt die niet meer dan kun je [hier \(Koppelingen naar een externe site.\)](#) lezen hoe je XAMPP kan installeren.

### Composer

Zorg er voor dat je [composer \(Koppelingen naar een externe site.\)](#) hebt geïnstalleerd. Composer is de installer van de WEB tools.

*Uitleg, wat is Composer?*

Als we een framework als Yii of Laravel gaan gebruiken dan installeren we eigenlijk een hele

grote doos met allemaal bouwblokken. Deze blokken moeten allemaal samen werken en zijn vaak afhankelijk van elkaar. De blokken worden door verschillende mensen ontwikkeld en op één of andere manier moet er voor worden gezorgd dat de juiste blokken zijn geïnstalleerd. Hiervoor dient Composer. Composer is als het ware de installer van PHP tools, deze worden vaak libraries of packages genoemd. Libraries zijn dan de blokken functionaliteit die je kunt gebruiken om een programma te maken.

<https://getcomposer.org/> (Koppelingen naar een externe site.)

Tip: Zorg dat XAMPP al is geïnstalleerd voordat je composer installeert; composer heeft namelijk de locatie van php.exe nodig.

In deze korte video wordt getoond hoe je Composer kunt installeren:

<https://www.youtube.com/embed/Rvb6DQ5PYHM>

## Create new yii Project

We maken het Yii project **world** aan.

Je hoeft niet zelf een map/folder aan te maken. Dit doet composer voor jou. Zorg ervoor dat je in de terminal in de parent-folder bent.

Vul de volgende code in de terminal:

```
composer create-project --prefer-dist yiisoft/yii2-app-basic world
```

Dit commando maakt een folder aan met de naam world. Dit is je nieuwe Yii project folder (er staan honderden bestanden in).

Let op dat het nieuwe Yii project wordt gemaakt in de folder waarin je de cmd box opent. Stel je wilt je jouw nieuwe project op je Desktop maken dan open je een cmd box en je tikt het commando `cd Desktop` in. Voer dan het composer commando uit (zoals hierboven aangegeven) en het nieuwe Yii project zal op je Desktop worden gemaakt.

## Start Development Server

yii heeft een ingebouwde php server als je die opstart dan draait jouw website op

<http://localhost:8080/> (Koppelingen naar een externe site.)

Let wel dat je XAMPP ook blijft draaien voor je database.

In VSC open je jouw Yii project en daarna open je een nieuwe terminal.

image-1618344180761.png

In de terminal type je dan het commando in:

```
php yii serve
```

De Yii webserver draait nu. Ga naar je browser en controleer of op je via localhost:8080 jouw nieuwe Yii project ziet.

Telkens als je aan de slag gaat met Yii, start je de Yii-webserver weer op met het commando [php yii serve](#)

Wil je cmd box gebruiken of lukt dit niet. Kijk dan naar de [instructiefilm \(Koppelingen naar een externe site.\)](#)

<https://www.youtube.com/embed/5idkjTWIL7g>

## Web.php

We moeten nu nog het een en ander configureren in Yii. We willen dat Yii zoveel mogelijk voor ons regelt. Om te beginnen mag Yii de routing regelen (dit wordt uitgelegd in de [volgende les](#) - [\(Koppelingen naar een externe site.\)](#)).

Open **config/web.php** - en zet het gedeelte dat urlmanager heet, uit commentaar (uncomment prettyURL). Het moet er als volgt uitzien (ongeveer op regel 50 van config/web.php).

```
'urlManager' => [  
    'enablePrettyUrl' => true,  
    'showScriptName' => false,  
    'rules' => [  
        ],  
    ],  
],
```

Dit is een instelling die de [routing \(Koppelingen naar een externe site.\)](#) (waarover later meer) eenvoudiger maakt.

## Database

Let er op dat je **database goed is gedefinieerd** en dat je ook de relaties heb vastgelegd. Dat kun je doen met [phpMyAdmin \(Koppelingen naar een externe site.\)](#). De ingebouwde [Yii CRUD-generator \(Koppelingen naar een externe site.\)](#) werkt alleen goed als de database en de relaties goed zijn vastgelegd.

## Database Installeren

Maak de database world en importeer deze file: <http://www.roc.ovh/attachments/22>

Lukt niet of weet je niet precies hoe je een database moet importeren? In [deze les \(Koppelingen naar een externe site.\)](#) staat beschreven hoe je deze database kunt installeren. In deze les staat de [SQL-file world.sql \(Koppelingen naar een externe site.\)](#) waarmee je de database World kan maken.

We gaan Yii vertellen welke database er moet worden gebruikt.

We openen de file **config/db.php** en zetten daar het volgende in:

```
return [  
    'class' => 'yii\db\Connection',  
    'dsn' => 'mysql:host=127.0.0.1;dbname=world',  
    'username' => 'root',  
    'password' => '',  
    'charset' => 'utf8',  
];
```

Let op dat je geen localhost gebruikt om naar je database te verwijzen. Gebruik 127.0.0.1.

Klaar? Test jouw installatie!

Je had al gezien dat je een leeg Yii project hebt, ga nu naar: <http://localhost:8080/gii>

Jouw scherm ziet er ongeveer zo uit:

[image.1665512780320.png](#)

Maak een screen shot van jouw hele scherm met browser en lever deze in.



# 03 Eerste CRUD

Je hebt Yii geïnstalleerd en je hebt de database *world* geïnstalleerd?

Het is tijd om een CRUD te maken met Yii.

## Eerste CRUD

We gaan een CRUD maken en met Yii is dat heel makkelijk.

We moeten eerst een verbinding met de database maken. Weet je nog welke letter van MVC daar ook alweer voor stond?

Yep, de M van Model. Het model beschrijft hoe jouw Yii applicatie een verbinding met de database kan maken. We gaan dus een Model maken.

### Stap 1. Model maken

Allereerst vergeet niet de yii server op te starten.

```
php yii serve
```

We gaan opnieuw naar de component builder, open <http://localhost:8080/gii/model> (Koppelingen naar een externe site.)

Klik links op 'Model Generator' en type de naam van de databasetabel **country** in:

image1594285274135.png

De naam van het model wordt automatisch aangemaakt **Country**. Let op dat de 'Model Class Name' met een hoofdletter begint; kijk goed naar het voorbeeld. De naam van een class (oop) wordt altijd met een hoofdletter geschreven.

Druk onderaan op de pagina op *preview*, Yii laat nu zien welke file hij gaan aanmaken voor je. Druk op **generate** om de file te maken.

De file **model/country** is nu aangemaakt, we gaan later meer in op de inhoud van deze file. Wat voor nu belangrijk is om te onthouden is dat de model/country.php file de verbinding is tussen de tabel country en Yii. Elke keer als Yii informatie uit de country tabel nodig heeft dan gebruikt Yii de model/country.php file

Omdat er vanuit de country tabel wordt verwezen naar de andere twee tabellen, **City** en **Countrylanguage** maak je nog twee modellen van deze twee tabellen.

In je model directory (folder) heb je nu dus de volgende files staan:

```
City.php
ContactForm.php
Country.php
Countrylanguage.php
LoginForm.php
User.php
```

## 2. CRUD compleet maken met controllers en views

Er zit een fout een versie 2.0.43 van het Yii framework (als je `php yii` in de terminal intypt dan verschijnt er vele tekst, maar helemaal als eerste verschijnt de versie). Heb je versie 2.0.43, check dan deze pagina:

<https://www.roc.ovh/books/yii/page/version-2043-crud-generator-issue>

Ga terug naar de component builder, [open \(Koppelingen naar een externe site.\)](#)

<http://localhost:8080/gii/model> (Koppelingen naar een externe site.)

Klik links op 'CRUD Generator' en type het volgende in:

image1618227641051.png

Let goed op de hoofdletters, deze zijn belangrijk. Het View path wordt leeg gelaten, daar kan Yii zelf iets voor kiezen.

Druk op preview, je ziet 8 files die klaar staan om te worden aangemaakt.

image1618227971123.png

Druk op **generate** om de files aan te maken.

### Controller

In de directory controllers staat een file CountryController.php. Dit is de controller van de country CRUD. Al het denkwerk dat nodig is om de country CRUD aan te sturen zit in de CountryController.php file.

## View

In de views/country directory staan de views. De index.php is de standaard view en deze file bevat de (voornamelijk HTML code) om de pagina weer te geven.

Weten we nog waar de controller en view ook alweer vandaan komen? Het Framework is gebouwd volgens de MVC-architectuur. In stap 1 hebben we het model gemaakt en in deze 2de stap hebben we de view en controller gemaakt.

image1612557541726.png

(deze sheet is uit de vorige les)

De controller is waar alle verzoeken van de gebruiker naar toe gaan. De view is de presentatielaag met voornamelijk HTML, CSS, en JavaScript. En in het model wordt de verbinding met de database gemaakt.

## Klaar

Ga naar <http://localhost:8080/country> (Koppelingen naar een externe site.)

image1594288457454.png

Je hebt een CRUD van Country gemaakt, compleet met search en sort-functionaliteiten. Je hebt ook een mooie pagina selector waarmee je daar de verschillende pagina's kan navigeren. Je kunt country's lezen/tonen, aanpassen, verwijderen en aanpassen. Probeer het maar! Wees niet bang om de database 'kapot' te maken want je kunt in een paar tellen de database opnieuw importeren.

In de volgende lessen gaan we de automatisch gebouwde applicatie stap-voor-stap aanpassen.

Voor nu kan je trots zijn op je eerste (?) web CRUD applicatie.

## Extra hulp met Filmpje

Niet gelukt? Kijk in dit filmpje nog een keer alle stappen door: [Instructie Yii - les1 \(Koppelingen naar een externe site.\)](#)

<https://www.youtube.com/embed/6NRMajgSmLw>

# Opdracht


**Maak voor de tabellen:**

- city
- countrylanguage

ook een CRUD met de Crud Generator

Controleer op <http://localhost:8080/city> (Koppelingen naar een externe site.) en <http://localhost:8080/country-language> (Koppelingen naar een externe site.) of de CRUD werkt.

# Opdracht

1. Ga in de code-editor naar de map **views** en probeer te vinden waar het overzicht Countries is gemaakt. Haal code weg zodat de kolom *Surface Area* niet meer wordt getoond.  

2. Kun je de in dezelfde view de groene knop 'Create Country' verplaatsen en onderaan de pagina plaatsen?  
Dus het moet er zo uit te komen zien:



# Inleveren

1. Lever een compleet screenshot in van de **country** read page (Grid View; zoals plaatsje hierboven met het rode kruis).
2. Lever een compleet screenshot in van de read page (Grid View) van **city**
3. Lever een compleet screenshot in van de read page (Grid View) van **countrylanguage**
4. Lever het (aangepaste) bestand *views/country/index.php* in - noem dit bestand jouw-naam-index.php

--

# 04 Routing

*In deze les leer je wat routing is en hoe je binnen Yii de routing kunt instellen.*

*Het begrip routing komt terug in alle of veel frameworks waarmee je web applicaties kan maken; Yii, Laravel, Flask, Django,...*

## Wat is routing? (Algemeen)

Routing is de manier waarop de webserver en jouw programma weet welke pagina er moet worden getoond. Kijk maar eens goed naar de URL in je browser. Je kunt de URL opdelen in 5 stukjes:

image-1594292626981.png

| Onderdeel   | Wat is het?                 | Uitleg   |
|-------------|-----------------------------|--|
| Scheme      | protocol                    | hoe wordt de informatie verzonden (http, https, ftp, smb, file, ...)   |
| domain name | domein                      | domein verwijst naar een server of een groep servers.  |
| file path   | path                        | verijst naar de file die de webserver moet 'uitvoeren'   |
| parameters  | variabelen                  | hiermee kun je informatie naar de webserver sturen.  |
| anchor      | verwijzing binnen de pagina | hiermee kun je naar een onderdeel van de pagina springen.<br>Voorbeeld:<br><a href="https://www.roc.ovh/link/302#bkmrk-nog-een-keer-in-sche">https://www.roc.ovh/link/302#bkmrk-nog-een-keer-in-sche</a> |

Het schema bepaald hoe de informatie wordt opgevraagd (bij een webserver is dit meestal http en https). De domain name is de naam van de server(s). Het file path verteld de server welke informatie er moet worden getoond. De parameters zijn variabele die mee kunnen worden gegeven bijvoorbeeld om waarden die een gebruiker heeft ingevoerd naar de webserver te sturen. Tot slotte is het anchor (weinig gebruikt) een verwijzing naar een stukje binnen een pagina.

Normaal gesproken komt het file path overeen met de locatie van de bestanden op de server. Het file path is het path vanaf de document root. Stel de document root staat op jouw laptop naar:

c:\www

Je hebt XAMPP opgestart en je gaat naar:

<http://localhost/opgave1/uitslag/index.php>

Dan wordt de file

c:\www\opgave1\uitslag\index.php

door de webserver uitgevoerd. Het file path wordt dus achter de *document root* geplaatst

## Hoe kom je van de URL in je browser naar de file op jouw laptop?

image\_1613336806883.png

Met routing zorg je ervoor dat je ergens een vertaling wordt gemaakt tussen het URL en de file PHP die moet worden uitgevoerd.

Alles wat hierboven is beschreven is de *standaard* routing. Zo werkt het op jouw laptop als je plain PHP gebruikt (dat is PHP zonder framework).

Routing zorgt ervoor dat het path en eventueel de parameters de juiste code uitvoeren. Bij eenvoudige web applicaties is het path gewoon de locatie van de file op de harddisk, maar voor meer ingewikkelde programma's kan dat al snel onoverzichtelijk worden.

Met een framework werkt het daarom anders.

## Routing in Yii

Standaard hoef je niets te doen voor routing. Er zit al heel veel voorgebakken in Yii.

Stel je typt in je browser in

localhost:8080/country/index

Dit path bestaat uit twee delen; **country** en **index**.

**country** verwijst naar controllers/CountryController en

**index** verwijst naar de public function action**Index** in de class **CountryController**.

Het path bestaat in Yii uit twee delen, de verwijzing naar de controller en de verwijzing naar de functie/method.

De vertaling van path naar controller en function/method gaat als volgt:

1. het eerste deel van het path is de controller-naam, maar
  1. de controller-naam begint altijd met een hoofdletter, en;
  2. als er een streepje (-) in het path staat dan is dat een nieuw woord en in de controller begint dit nieuwe woord met een hoofdletter. Voorbeeld: `dit-is-het-eerste-deel` wordt `DitIsHetEersteDeel`
2. Het tweede deel van het path wordt opdezelfde manier vertaald maar er wordt ook nog eens het woord action voorgezet. Voorbeeld: `dit-is-het-tweede-deel` wordt `actionDitIsHetTweedeDeel`

Een paar voorbeelden:

| path                    | file                 | public function |
|-------------------------|----------------------|-----------------|
| /kaart/overzicht        | KaartController      | actionOverzicht |
| /klas-lokaal/overzicht  | klasLokaalController | actionOverzicht |
| /klas-lokaal/stoel-maat | klasLokaalController | actionStoelMaat |
| /klas-lokaal/index      | klasLokaalController | actionIndex     |
| /klas-lokaal            | klasLokaalController | actionIndex     |

In de laatste regel ontbreekt het tweede deel van het path. In dat geval wordt de actionIndex uitgevoerd.

## Oefening

We gaan in onze World-applicatie oefenen met routing

(1) Zet in de CountryController in de CountryController class een nieuwe public function:

```
public function actionHello() {  
    echo "Hello World!";  
    exit;  
}
```

**Opdracht:** Save de aangepaste CountryController en bedenk met welke url je nu deze nieuwe functie (method) moet aanroepen.

Controleer dat, zodat je in jouw browser het volgende ziet:

image-1594296881002.png

(2) Maak een nieuwe file **controllers/ExampleController.php** en zet daar het volgende in.

```
<?php  
  
namespace app\controllers;  
  
use Yii;  
use yii\web\Controller;  
  
/**  
 * CountryController implements the CRUD actions for country model.  
 * Code by Navneet Shiravadakar  
 */  
  
class ExampleController extends Controller  
{  
    public function actionSay($message = 'empty')  
    {  
        echo "Hello $message";  
        exit;  
    }  
}
```

**Opdracht:**

Vervang de naam *Navneet Shiravadakar* in het commentaar door jouw naam.



Ga nu naar <http://localhost:8080/.....?message=friend> maar vervang de puntjes zodat je de functie (method) actionSay van de ExampleController uit laat voeren.

**Opdracht:** Pas de parameter in de url aan zodat er op het scherm komt "Hello" gevolgd door jouw voornaam. Gebruik voor jouw naam één woord.

## Inleveren

1. Het bestand: controllers/ExampleController.php
2. Screenshot van het volledig scherm met browser waarop Hello gevolgd door jouw naam op het scherm verschijnt.

--

## *Verdieping (manual routing)*

In de [vorige les](#) hebben een instelling in de config/web.php aangepast. Hierdoor hebben we eenvoudige routing aan gezet. Als je dit uit zet dan werkt de routing iets anders. Voor nu niet belangrijk maar als je wilt weten hoe het precies zit, kijk dan naar dit Engelse Youtube filmpje:

<https://www.youtube.com/embed/QcZiS43iVxU>

Verder is er nog meer informatie te vinden op de officiële Yii documentatie pagina's:

<https://www.yiiframework.com/doc/guide/2.0/en/runtime-routing>

--

# 05 Menu

*In deze les gaan we routing zoals we dat in de vorige les hebben geleerd toepassen. We gaan een menu maken en we gaan de menu's koppelen aan de routing*

## Views

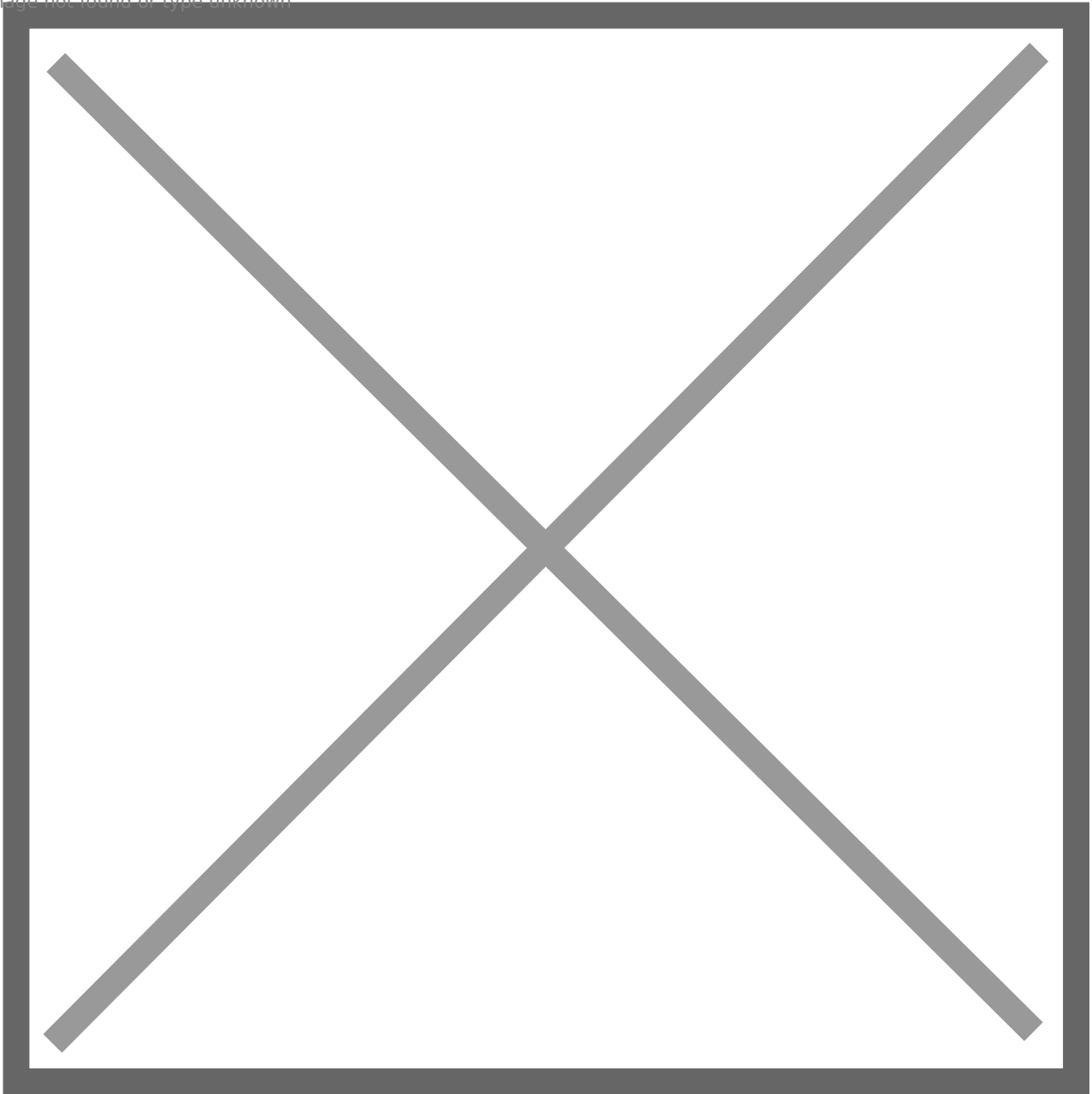
In de folder views zien we drie folder staan:

1. **country**  
In de country folder staan de views die we met de CRUD-generator hebben gegeneerd.
2. **lay-outs,**  
hierin staat de standaard lay-out van elke Yii pagina.
3. **site**  
hierin staan een aantal standaard Yii pagina's zoals de *about* en de *contact* page.

Elke pagina die je laadt met Yii wordt gevormd door main.php in de lay-outs view.

In de main.php staat `<?= $content ?>` dit is kort voor `<?php echo $content ?>` . Het toont jouw pagina die in de variabele \$content staat:

Image not found or type unknown



## *Nav Widget in main.php*

In de main.php in de folder lay-outs staat de standaard Yii pagina structuur. Het menu is hier ook een onderdeel van.

In de main.php staat een stuk PHP-code die gebruik maakt van een Yii widget. Een Yii widget is eigenlijk een plug-in zoals we die ook bij WordPress kennen.

De code voor de menu bar heeft de volgende structuur:

```
<?php
```

[/] plaats deze code in de main.php en vervang daarmee de standaard menu

```
NavBar::begin([

    // hier wordt het type en de stijl van de menu bepaald
    'brandLabel' => Yii::$app->name, // de naam van het menu
    'brandUrl' => Yii::$app->homeUrl, // de home page waar je naar toe gaat als je op de naam klikt
    'options' => [
        // 'class' => 'navbar-inverse navbar-fixed-top', // de Bootstrap 4 style van het menu
        'class' => 'navbar-expand-md navbar-dark bg-dark fixed-top',
    ],

]);

echo Nav::widget([

    // hier worden de menu's en menu items bepaald
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => [
        [ 'label' => 'Country',
            'items' => [
                ['label' => 'Overzicht', 'url' => ['/country/index', '']],
                ['label' => 'Voeg toe', 'url' => ['/country/index', '']],
                ['label' => 'Europa', 'url' => ['/country/index', '']],
            ],
        ],
    ],
]);

NavBar::end();
?>
```

In Yii worden parameters voor Widget via associative arrays mee gegeven. Deze hebben de structuur:

```
[ 'key1' => 'waarde1', 'key2' => 'waarde2' ]
```

de waarde kan dan ook weer een associative array zijn:

```
[ 'key1' => 'waarde1', 'key2' => [ 'andere_key1' => 'andere_waarde1', ..... ] ]
```

# Opgaven

1. Pas het menu aan en plaats de code zoals hierboven beschreven:

image1613993167595.png

2. Om een country toe te voegen moet in de controller *CountryController* de method/function *actionCreate()* worden uitgevoerd.  
Bedenk hoe je de routing naar deze pagina opzet en pas het menu aan 'voeg toe' zodat deze is gekoppeld aan *actionCreate()* van de *CountryController*.

3. In elk menu-item staat de route, bijvoorbeeld 'country/index' en daarna staat een ''. Deze laatste lege string zijn de parameters. Als we alle landen uit Europa willen filteren dan kunnen we de volgende URL gebruiken:

image1613993783698.png

Het gedeelte na de ? is de parameter. In Yii zet je een paramter als volgt in een associatieve array (element): 'countrySearch[Continent]'=>'Europe'

Het hele menu-item ziet er dus als volgt uit:

image1615820099350.png

----- Pas je menu aan en test uit of alle drie de menu-items goed werken.

4. Maak nu een extra menu-item en noem dat Asia. Dit menu-item laat alle landen van Asia zien.
5. Maak nu een menu-item voor alle werelddelen: North America, South America, Asia, Europe, Africa, Antarctica en Oceania.
6. Maak een extra menu city met als items 'Overzicht' en 'Voeg toe'.

image1614009669414.png

# Inleveren

Lever jouw views/layouts/main.php file in. Hierin staat als het goed is jouw menu.

# 06 Grid View

*Dit In deze les leer je de basis aanpassingen in de Gridview maken.*

*Je leert hoe je kolommen wel of niet **zichtbaar** kan maken, hoe je de kolom **headers** kan aanpassen, hoe je de kolom **breedte** kan aanpassen, hoe je de **inhoud** van de kolom zelf kan aanpassen en als laatste leren hoe we een **link** kunnen maken van een kolom.*

## Country overzicht aanpassen

Onze CRUD-generator (Gii) heeft voor ons een country overzicht gemaakt. Als je de CountryController.php opent dan zie je dat function (method) actionIndex de view index aanroept. Open deze index file. Deze staat in views/country.

## Gridview Widget

Je ziet dat er op regel 23 een zogenaamde widget wordt aangeroepen.

Een Widget is een plug-in waarmee je functionaliteit aan Yii toevoegt. De Gridview Widget is misschien wel de meest uitgebreide widget en het zorgt ervoor dat we een mooi overzicht van onze data krijgen.

### 1. Kolommen aanpassen

In de parameters van deze widget zie je dat de kolommen *Code*, *Name*, *Continent*, *Region* en *SurfaceArea* worden afgedrukt. Je ziet ook dat bepaalde kolommen niet worden afgedrukt deze staan in commentaar.

## Opdracht 1

Verander de lay-out in het country overzicht en laat de volgende kolommen in deze volgorde zien:

*Code, Name, Population, SurfaceArea en Capital.*

*Population* laat inwoneraantal zien en *SurfaceArea* laat oppervlakte zien.

## 2. Kopjes aanpassen

Stel we willen het kopje boven de kolom veranderen, dan kan dan als volgt. Vervang bijvoorbeeld 'SurfaceArea' in de GridView door het volgende array.

```
[ 'label' => 'Oppervlakte',  
  'attribute' => 'SurfaceArea',  
],
```

Je ziet dat we de surface area nu hebben vervangen door een associative array. Dus waar in de code het woord 'SurfaceArea' staat plaats je het associative array zoals hierboven is aangegeven.

Dit verteld de GridView snippet dat het label Oppervlakte moet zijn (dit wordt boven aan de pagina gezet) en dat de waarde in de kolom moet worden gevuld met 'SurfaceArea'.

## Opdracht 2

Pas nu de kolom *Population* aan; en zet in de kolom header 'Inwoners'.

## 3. Kolom breedte aanpassen

We gaan de kolombreedte aanpassen. Vervang het blok van net door het volgende:

```
[ 'label' => 'Oppervlakte',  
  'attribute' => 'SurfaceArea',  
  'contentOptions' => ['style' => 'width:30px; white-space: normal;'],  
],
```

## Opdracht 3

Pas de breedte van de kolom *Inwoners (Population)* ook aan; zet de breedte op 30 pixels.

## 4. Waarde van kolom aanpassen

Stel we willen km2 achter de oppervlakte plaatsen, dan kunnen we de waarde van de kolom als volgt aanpassen:

```
[
  ['label' => 'Oppervlakte',
    'attribute' => 'SurfaceArea',
    'format' => 'raw',
    'value' => function($data) {
      return sprintf("%8d k&#13217", $data->SurfaceArea);
    }
  ]
]
```

Regel 4 t/m 7 in deze code verandert de waarde van de kolom.

## Opdracht 4

Test uit wat de code doet, haal 'format'=> 'raw' weg en kijk wat er gebeurt.

Waarom is deze regel nodig?

Vervang regel 6 door

```
return 'hallo';
```

en kijk wat er gebeurt.

## 5. Link maken van een kolom

In de *Capital* kolom zien we geen plaatsnaam, maar een nummer. Dit is het ID van een plaats in de *city* tabel.

Dus de *Capital* kolom uit *country* is de foreign key die wijst naar de primary key *ID* in *city*.

In de *city* view kun je een plaats naam zoeken door bijvoorbeeld de volgende URL te gebruiken:

[http://localhost:8080/city/index?CitySearch\[ID\]=179](http://localhost:8080/city/index?CitySearch[ID]=179)



179 is het ID van Brussels, de hoofdstad van België.

We gaan nu de value van de kolom Capital vervangen door de link.

Laten we om te beginnen de kolom *Capital* een beetje gaan aanpassen:

## Opdracht 5

Vervang de kolomnaam van de kolom Capital door 'Hoofdstad'.

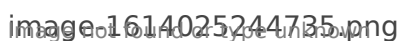
Verander de breedte van deze kolom naar 30px.

Vervang de value van de kolom in

```
'value' => function($data) {  
    return "/city/index?CitySearch[ID]=179";  
}
```

(alle kolommen krijgen dus dezelfde waarde).

Je ziet nu:



Dat is nog geen link. Dat komt omdat we met `<a href="">...</a>` een link moeten maken. Dat kan maar dat is lastig omdat we met `"` in `"` quotes zitten. Dus dubbele quotes in dubbele quotes. Dat kan, maar het is een gepriegel.

Met de Yii module Html kunnen we dit eenvoudiger doen.

Zet eerst boven aan in de code:

```
use yii\helpers\Html;
```

En we vervangen nu de regel die de waarde returned in:



We zijn er bijna, we moeten nu alleen de 179 nog variabel maken.



## Opdracht 6

Maak je code werkend met bovenstaande code.

Als je nu op het nummer klikt ga je naar de hoofdstad van het betreffende land.

Verander nu de tekst van de link zodat je in de kolom Hoofdstad, in 'naar hoofdstad'.

Als je er op klikt dan ga je dus naar de index view van de hoofdstad.

Verander ook de breedte van de kolom naar 120px zodat de kolom netjes past.

De kolom ziet er dus als volgt uit:

image-1614026822965.png

## Opdracht 7

(a) Zet de kolommen in de volgende volgorde:

Code - Name - Hoofdstad - Population - Oppervlakte

(b) Verander de kolomnamen als volgt:

Code - Naam - Hoofdstad - Inwoners - Oppervlakte

(c) Zorg dat de naam van het land vet gedrukt wordt (bold).

(d) Haal vervolgens de volgnummers (1,2,3,4....) weg en verplaats de edit kolom (de laatste) naar voren.

Het hele overzicht ziet er uit zoals te zien is op het volgende plaatje:

image-1617127616569.png

Dit is het eindresultaat.

## Opdracht 8

De bevolkingsdichtheid zegt iets over de hoeveelheid inwoners in een land. Natuurlijk heeft Amerika meer inwoners dan Nederland, maar het land is ook vele groter. Om te vergelijken hoe dichtbevolkt een land is, kun je uitrekenen hoeveel mensen er gemiddeld per km<sup>2</sup> wonen.

bevolkingsdichtheid = aantal inwoners / oppervlakte.

Stel een land heeft een oppervlakte van 100 km<sup>2</sup> en je er wonen 200 mensen. Dan heb je in dit land dus gemiddeld 2 inwoners per km<sup>2</sup>, want:

$$2 = 200 / 100$$

Maak nu een nieuwe kolom in het *country* overzicht en bereken daarin de bevolkingsdichtheid.

## Inleveren

Voer alle opdrachten uit.









Aan het einde heb je jouw gridview aangepast en ziet het er zo uit:

Home / Countries

Countries

Showing 1-20 of 239 items.

VB

|  | Code                 | Naam                 | Hoofdstad                      | Inwoners             | Oppervlakte             | Bevolkingsdichtheid |
|--|----------------------|----------------------|--------------------------------|----------------------|-------------------------|---------------------|
|  | <input type="text"/> | <input type="text"/> | <input type="text"/>           | <input type="text"/> | <input type="text"/>    |                     |
| <br> | ABW                  | Aruba                | <a href="#">naar hoofdstad</a> | 103000               | 193 km <sup>2</sup>     | 534                 |
| <br> | AFG                  | Afghanistan          | <a href="#">naar hoofdstad</a> | 22720000             | 652090 km <sup>2</sup>  | 35                  |
| <br> | AGO                  | Angola               | <a href="#">naar hoofdstad</a> | 12878000             | 1246700 km <sup>2</sup> | 10                  |
| <br> | AIA                  | Anguilla             | <a href="#">naar hoofdstad</a> | 8000                 | 96 km <sup>2</sup>      | 83                  |

## Inleveren

1. Maak een schermafdruck van jouw Countries aangepaste gridview en lever deze in.
2. Stuur ook jouw (aangepaste) country view in.

# 07 En verder?

We hebben kennis gemaakt met ons eerste PHP Framework.

We hebben heel veel geleerd:

- wat CRUD is
- wat MVC is
- installatie van Yii
- maken van een CRUD in Yii
- wat is routing en hoe het werkt
- aanpassen grid view in Yii
- aanpassen van menu in Yii

We hebben een mooie basis gelegd voor volgend jaar. Mocht je zelf vast verder willen dan staat (een vorige versie) van deze lessen op: <https://www.roc.ovh/books/yii>

We sluiten af met een laatste quiz.