Yii Introduction

1 Intro - Wat is Yii?

datasource

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).

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 verzamelingen 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 alles 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 is minder flexibel en om het precies zo te krijgen 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 projecten en examens 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.image.png

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

MVC

MVC, wat is dat? MVC staat voor **Model**, **View** en **Controller**. 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.

https://www.youtube.com/watch?v=FCkDEHWDATI

https://www.youtube.com/embed/FCkDEHWDATI

CRUD?

Nog even....waar staat CRUD ook alweer voor? CRUD staat voor**:**

Create	Nieuwe toevoegen
Read	Bekijken / lezen
Update	Bewerken
Delete	Verwijderen

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.

? Quizvragen

1. **Wat is een framework en waarom zou je het gebruiken bij het maken van een webapplicatie?**
   Leg in je eigen woorden uit wat een framework is en wat het voordeel is ten opzichte van alles zelf programmeren.
2. **Wat zijn volgens de tekst belangrijke vragen die je moet stellen voordat je kiest voor een bepaald framework?**
   Noem minstens twee overwegingen en leg kort uit waarom die belangrijk zijn.
3. **Wat is volgens de tekst het verschil tussen Laravel en Yii?**
   Leg uit in welk opzicht Yii sneller is dan Laravel, maar ook waar je tegenaan zou kunnen lopen.
4. **Wat betekenen de letters M, V en C in MVC**
   Beschrijf per onderdeel kort de rol die het speelt binnen een webapplicatie.
5. **Hoe werken Model, View en Controller samen?**
   Beschrijf het proces dat plaatsvindt als een gebruiker op een knop drukt.

6. **Wat betekent CRUD en wat moet je hierover kunnen voor het examen?**
   Noem alle vier onderdelen van CRUD en leg uit waarom dit belangrijk is voor je examenproject.
7. **Waarom is Yii een goede keuze voor het examenproject?**
   Noem minstens twee redenen waarom Yii goed past bij de opdrachten die je moet maken.

? Inleveren

- Beantwoord de vragen en lever dit in als .txt of .pdf-bestand.
  (llet op dit is een goede voorbereiding op de kennis-check)

2 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 (of een andere tool) een database kunt maken en hoe je een .sql file kan importeren. Ook heb je XAMPP ge nstalleerd.

?Installatie

De installatie kent een aantal stappen. Hieronder zijn de 5 stappen beschreven die je moet uitvoeren om tot een succesvolle Yii installatie te komen.

?stap 1 - XAMPP

Je hebt XAMPP ge nstalleerd. 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 lezen hoe je XAMPP kan installeren.

?stap 2 -Composer

Zorg er voor dat je composer hebt ge nstalleerd. 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 n of andere manier moet er voor worden gezorgd dat de juisten blokken zijn ge nstalleerd. Hiervoor dient Composer. Composer is als het ware de installer van PHP tools, deze worden vaak libraries of packages genoemd. Librabries zijn dan de blokken functionaliteit die je kunt gebruiken om een programma te maken.

https://getcomposer.org/

Let op: check goed of de installatie errors/warnings geeft. Zijn er errors, lees deze dan goed en los de issues met behulp van ChatGPT op!

Tip: Zorg dat XAMPP al in ge nstalleerd 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

?stap 3 - 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 (in de folder/map waar je nu staat) 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).

??Error, warning ?

image.png	Let op, check of je **errors** en of **warnings** krijgt! Zie je die, probeer die dan met behulp van ChatGPT op te lossen! Extra uitleg: bij sommige installaties van XAMPP staan bepaalde extensies niet aan. Zo staat bijvoorbeeld de ZIP extensie soms uit in de PHP config (XAMPP, php.ini). Vraag chatgpt om dit op te lossen. Als dit zo is dan kan de Yii installatie fout gaan. Check daarom de output en pas de configuratie met behulp van ChatGPT aan. Als je opnieuw wilt beginnen moet je eerst je oude foutieve installatie verwijderen (in dit geval de map "*world*"). Doe dit vanuit de file explorer.

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.

?stap 4 - Start Development Server

yii heeft een ingebouwde php server als je die opstart dan draait jouw website op http://localhost:8080/

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.

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

?stap 5 - Configure 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).

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 (waarover later meer) eenvoudiger maakt.

?stap 6 - Database

Let er op dat je **database goed is gedefinieerd** en dat je ook de relaties heb vastgelegd. Dat kun je doen met phpMyAdmin . De ingebouwde Yii CRUD-generator werkt alleen goed als de database en de relaties goed zijn vastgelegd.

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

Lukt niet of weet je niet precies hoe je een database moet importeren? In deze les staat beschreven hoe je deze database kunt installeren. In deze les staat de SQL-file world.sql 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.

?stap 7- Klaar?

Klaar? Dan moeten we nog even testen of alles werkt.

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:

Screenshot 2022-05-20 183633-1.png

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

? Inleveren

1. Screenshot van Yii waarin je http://localhost:8080/gii laat zien. Dit is de Model en Crud generator die we in de voglende stap gaan gebruiken.

3 CRUD aanmaken

? Leerdoelen

- Je kunt met de Gii CRUD Generator een complete CRUD maken voor een databasetabel.
- Je begrijpt dat je eerst een model moet genereren voordat je een CRUD kunt maken.

?CRUD

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

We moeten eerst een verbiding 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 - models maken

We gaan opnieuw naar de component builder, open http://localhost:8080/gii/model

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

image-1594285274135.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.

Controleer de model folder en als het klopt (let op de hoofdletters!) ga dan niet door.

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
```

? Stap 2 - CRUD maken

We gaan de complete CRUD maken.

Ga terug naar de component builder.

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

image-1618227641051.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.

image-1618227971123.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.

image-1612557541726.png

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

image-1594288457454.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

https://www.youtube.com/embed/6NRMaJgSmLw

? Uitleg

Je hebt al een CRUD gemaakt voor de tabel `country`. Nu ga je hetzelfde doen voor de tabellen `city` en `countrylanguage`.

Denk eraan: je moet eerst de modellen genereren voor deze twee tabellen met de Model Generator, voordat je de CRUD maakt.

?? Opdracht

1. Open de **Model Generator** op http://localhost:8080/gii/model
2. Maak het model voor de tabel `city`.
3. Maak het model voor de tabel `countrylanguage`.
4. Open daarna de **CRUD Generator** op http://localhost:8080/gii/crud
5. Genereer een CRUD voor `City`.
6. Genereer een CRUD voor `Countrylanguage`.

Let goed op de hoofdletters bij het invullen van de modelnamen!

Controleer op

1. http://localhost:8080/city
2. http://localhost:8080/countrylanguage

of de CRUD werkt.

GIT

"*Meester, gisteren werkje het nog en ik heb niets gedaan?*"

Bij Yii loop je soms vast. Je maakt wijzigen in een door het framework gemaakte files. Soms werkt er opeens *niets* meer (echt dat gebeurt iedereen!).

Met GIT kun je ervoor zorgen dat je steeds een werkende versie opslaat. Werkt het dan niet meer, dan kun je terug naar de vorige werkende versie.

Wert GIT nog niet in jouw VCS? Hier staat hoe je GIT in VCS kunt 'aanzetten'.

? Reflectie

- Waarom moet je eerst een model aanmaken voordat je een CRUD kunt genereren?
- In welke folder(s) worden er bestanden aangemaakt door de model-generator?
- In welke folder(s) worden er bestanden aangemaakt door de CRUD-generator?

? Inleveren

- Maak een screenshot van `http://localhost:8080/city` waarop de CRUD werkt.
- Maak een screenshot van `http://localhost:8080/countrylanguage` waarop de CRUD werkt.

4 View aanpassen

?? Opdracht - deel 1

Zorg eerst dat, nu alles nog werkt, je met GIT een commit uitvoert. Je kan dan altijd terug als je wat 'kappot' maakt.

Ga in de code-editor (VSC) 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.

Probeer maar wat, als je de boel 'kappot' maakt aan ga je met behulp van GIT naar de vorige versie.

image-1612558210976.png

?? Opdracht - deel 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:

image-1612558884647.png

? Resultaat

Jouw *country* overzicht ziet er dus zo uit:

image.png

? 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

5 Routing in Yii

? Leerdoelen

- Je weet wat routing is in het algemeen.
- Je begrijpt hoe routing werkt binnen Yii.
- Je kunt URL’s analyseren en begrijpen hoe ze naar controllers en acties verwijzen.

? Uitleg

Wat is routing?

Routing is de manier waarop een webserver en je programma bepalen welke pagina moet worden. Kijk maar eens goed naar de URL in je browser: deze bestaat uit verschillende onderdelen.

image-1594292626981.png

Onderdeel	Wat is het?	Uitleg
Scheme	Protocol	Bepaalt hoe de informatie wordt verzonden (zoals http, https, ftp...)
Domeinnaam	Domein	Verwijst naar een server of groep servers.
Path	Bestandspad	Verwijst naar het bestand dat de webserver moet uitvoeren.
Parameters	Variabelen	Sturen informatie mee naar de server (zoals bij formulieren).
Anchor	Paginaanker	Springt naar een specifiek onderdeel binnen een pagina.

Voorbeeld-URL: `https://www.roc.ovh/link/302#bkmrk-nog-een-keer-in-sche`

Bij een simpele PHP-site zonder framework komt het pad overeen met de bestandslocatie op je computer.

> https://localhost/ **opgave1/uitslag/index.php**
>
> Dan wordt de file
>
> c:\xampp**htdocs\opgave1\uitslag\index.ph**

Wat is de document-root van de webserver?

De **document root** van de webserver is waar de webserver al zijn bestanden kan vinden en waar die heen gaat als je alleen de hostname gebruikt.

c:\xampp\htdocs wordt ook wel de docuemnt root van de webserver genoemd omdat:

https://localhost verwijst naar **c:\xampp\htdocs** (in een standaard geintalleerde XAMPP omgeving).

Routing bij frameworks zoals Yii

Frameworks zoals Yii gebruiken een ander soort routing. De URL wordt niet direct vertaald naar een bestand, maar naar een **controller** en een **actie** (methode).

Voorbeeld:

```
http://localhost:8080/country/index
```

De URL hierboven verwijst naar:

De controller	**country** in ⟶ `CountryController` in `controllers/`
De actie	**index** in ⟶ methode `actionIndex()` binnen die controller

De standaardregels in Yii:

- Het **eerste deel** van het pad wordt omgezet naar een **controller**. Elk woord begint met een hoofdletter. Een streepje (`-`) betekent een nieuw woord.
- Het **tweede deel** van het pad wordt een **methode (functie in de controller)**, voorafgegaan door `action`, en met dezelfde schrijfwijze als hierboven.

Voorbeelden

Pad	Controller	Actie
/kaart/**overzicht**	KaartController	**actionOverzicht**()
/klas-lokaal/**overzicht**	KlasLokaalController	**actionOverzicht**()
/klas-lokaal/s**toel-maat**	KlasLokaalController	**actionStoelMaat**()
/klas-lokaal/**index**	KlasLokaalController	**actionIndex**()
/klas-lokaal	KlasLokaalController	**actionIndex**()

Als het tweede deel ontbreekt, wordt automatisch `actionIndex()` gebruikt.

?? Opdracht – Hello World!

We gaan in onze World-applicatie oefenen met routing.

1. Open `CountryController.php` in de map `controllers`.
2. Voeg daarin de volgende methode toe:

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

3. Sla het bestand op.
4. Bedenk: welke URL moet je nu gebruiken om deze functie aan te roepen?

5. Open je browser en test de URL. Als het goed is zie je:

   image-1594296881002.png

? Controle

- Is de tekst `Hello World!` zichtbaar in je browser?

? Inleveren

- Maak een screenshot van je browser waarop **Hello World!** zichtbaar is en waarin tevens de URL zichtbaar is.

Routing met een parameter

We gaan een nieuwe controller maken en oefenen met routing waarbij een **parameter** wordt doorgegeven via de URL.

1. Maak een nieuwe file aan in de map `controllers` met de naam `ExampleController.php`.
2. Plaats hierin de volgende code, maar **vervang de naam Ayoub op beide plekken door jouw eigen naam**:

```
<?php

namespace app\controllers;

use Yii;
use yii\web\Controller;

/**
 * ExampleController – voorbeeldcode voor routing.
 * Code door [Jouw Naam]
 */
class ExampleController extends Controller
{
  public function actionSay($message = '')
  {
    echo "Hello $message";
```

```
    exit;
  }
}
```

3. Sla het bestand op.
4. Open je browser en ga naar de juiste URL om `actionSay()` uit te voeren. Je gebruikt hiervoor:

```
http://localhost:8080/example/say
```

Je ziet nu als het goed is: **Hello**

Kun je via de URL ook een **parameter** meegeven, zodat in plaats van "Hello" er bijvoorbeeld "Hello Max" staat?

Hint: voeg je naam toe aan de URL als volgt:

```
http://localhost:8080/example/say?message=Max
```

? Controle

- Wordt de pagina `Hello [jouw naam]` getoond in de browser?
- Werkt de parameter in de URL zonder dat je iets hoeft te veranderen aan de code?

? Reflectie

- Als je jouw naam in de URL zet wordt deze in de variabele `$message` gezet, er staat immers `echo "Hell $message";`
  Ho ekomt het dat jouw naam in de variable met de naam `$message` staat?

? Inleveren

- Het bestand: `controllers/ExampleController.php`
- Een screenshot van je volledige scherm met browser waarop de URL en **Hello [jouw naam]** te zien is.
- Een antwoord op de refelctievraag (in txt of pdf).

06 Menu

? Leerdoelen

- Je begrijpt hoe het menu (navigatie) in Yii is opgebouwd.
- Je kunt menu-items koppelen aan routes (routing).
- Je kunt submenu's maken en parameters gebruiken in je menu-items.

? Uitleg

In deze les ga je routing zoals we dat in de vorige les hebben geleerd toepassen. Je maakt een menu en koppelt de menu-items aan routes in je applicatie.

◫ Views folder

In de map `views` vind je drie mappen:

- **country:** bevat de views die je met de CRUD-generator hebt aangemaakt.
- **layouts:** bevat de standaard lay-out (zoals `main.php`) van elke pagina.
- **site:** bevat standaardpagina’s zoals “About” en “Contact”.

Elke pagina in Yii gebruikt de layout `main.php` en/of `_header.php`

◫ Experimenteer

Check waar het menu staat (main.php of _header.php)? Probeer iets te veranderen en kijke of je snapt hoe het menu in elkaar zit.

◫ Menu via Widget

In de layout-bestand `main.php` staat een stuk code dat het menu opbouwt met een zogenaamde **Widget**. Een widget is een soort herbruikbaar codeblok, vergelijkbaar met WordPress plugins.

Vervang de standaard **menucode** in `main.php` of in `_header` door deze aangepaste versie.

> Maar let op, deze code is oude code en werkt niet meer op nieuwe versies van Yii, check je `_header.php`. Je herkent daar het menu in en je kunt **delen** uit de volgende code overnemen om het menu in `_header` aan te passen.

```
<?php
use yii\bootstrap5\Nav;
use yii\bootstrap5\NavBar;

NavBar::begin([
  'brandLabel' => Yii::$app->name,
  'brandUrl' => Yii::$app->homeUrl,
  'options' => [
    'class' => 'navbar-expand-md navbar-dark bg-dark fixed-top',
  ],
]);

echo Nav::widget([
  'options' => ['class' => 'navbar-nav navbar-right'],

  // ***
  // Dit gedeelte moet je overnemen als je het menu moet integreren in _header.php
  // ***
  'items' => [
    [
      'label' => 'Country',
      'items' => [
        ['label' => 'Overzicht', 'url' => ['/country/index']],
        ['label' => 'Voeg toe', 'url' => ['/country/create']],
        ['label' => 'Europa', 'url' => ['/country/index', 'CountrySearch[Continent]' =>
'Europe']]
      ],
    ],
    [
      'label' => 'City',
      'items' => [
        ['label' => 'Overzicht', 'url' => ['/city/index']],
        ['label' => 'Voeg toe', 'url' => ['/city/create']],
      ],
    ],



  ],
]);
```

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

Zoals je ziet gebruiken we **associatieve arrays** zoals:

```
['label' => 'Europa', 'url' => ['/country/index', 'CountrySearch[Continent]' => 'Europe']]
```

?? Opdrachten

1. Vervang de standaard menucode in `main.php` of `_header.php` door het bovenstaande voorbeeld.
   image.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. Voeg het menu-item **Asia** toe, dat alleen landen van Azië toont.
4. Voeg vervolgens voor elk werelddeel een apart menu-item toe: Asia, Africa, North America, South America, Oceania en Antarctica.
5. Voeg een nieuw hoofdmenu toe: **City** met twee items: **Overzicht** en **Voeg toe.**

? Inleveren

- Lever het aangepaste bestand `views/layouts/main.php` of `views/layout/_header.php` in.

07 Gridview

In deze les leer je 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.

?? Opdracht

De opdracht bestaat uit 8 stappen. In deze stappen gaan we de country view aanpassen (views/country/index.php).
Test na elke stap of de aanpassingen werken!

◫ Stap 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.

- Verander de lay-out in het country overzicht en laat de volgende kolommen in deze volgorde zien:
  Code , *Name* , *Population* , *Surface Area* en *Capital* .
  Population laat inwoneraantal zien en *SurfaceArea* laat oppervlakte zien.

Test of het werkt!

◫ Stap 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 assiociative array. Dus waar in de code het woord 'SurfaceArea' staat plaats je het assiociative 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'.

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

Test of het werkt!

Stap 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;'],
],
```

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

Test of het werkt!

Stap 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 5 t/m 7 in deze code verandert de waarde van de kolom. Dit is een functie en de return waarde van de functie wordt in de kolom afgedrukt.

Test of het werkt!

Als je achter return bijvoorbeeld "Hallo" zet dan wordt er in de kolom Oppervlakte overal Hallo afgedrukt. Probeer maar!

Stap 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 plaatsnaam zoeken door bijvoorbeeld de volgende URL te gebruiken:

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

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

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:

1. Vervang de kolomnaam van de kolom Capital door 'Hoofdstad'.
2. Verander de breedte van deze kolom naar 30px.
3. Vervang de value van de kolom in `'format' => 'raw', 'value' => function($data) { return "/city/index?CitySearch[ID]=179"; }` (alle kolommen krijgen dus dezelfde waarde).

Je ziet nu (in de browser):

image-1614025244735.png

Test of dit werkt!

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

Met de Yii module Html kunnen we dit eenvoudiger doen.

We vervangen de regel die de waarde returned in:

image-1614026085274.png

Je ziet nu in de browser een nummer (FK naar *city*) en als je daar op klikt, ga je naar de city *Bruxelles [Brussel]*

Test of dit werkt!

We zijn er bijna, we moeten nu alleen de 179 nog variabel maken, anders linkt elke regel naar Brussel.

image-1614026443516.png

◻◻ Stap 6 - Testen en laatste aanpassing

1. Maak je code werkend met bovenstaande code.
2. Als je nu op het nummer klikt ga je naar de hoofdstad van het betreffende land.
   Test of dit werkt!
3. Verander nu de tekst van de link zodat je in de kolom Hoofdstad, in 'naar hoofdstad'.
4. Als je er op klikt dan ga je dus naar de index view van de hoofdstad.
5. Verander ook de breedte van de kolom naar 200px zodat de kolom netjes past.

De kolom ziet er dus als volgt uit:

image-1614026822965.png

◻◻ Stap 7, kolommen aanpassen

1. Zet de kolommen in de volgende volgorde:
   Code - Name - Hoofdstad - Inwoners- Oppervlakte
2. Verander de kolomnamen als volgt:
   Code - Naam - Hoofdstad - Inwoners - Oppervlakte
3. Zorg dat de naam van het land vet gedrukt wordt (bold).
4. Haal vervolgens de volgnummers (1,2,3,4....) in de eerste kolom weg
5. 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

◻◻ Stap 8 - bevolkingsdichtheid

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 km2 (dit een gebied van 1000x1000 meter) wonen.

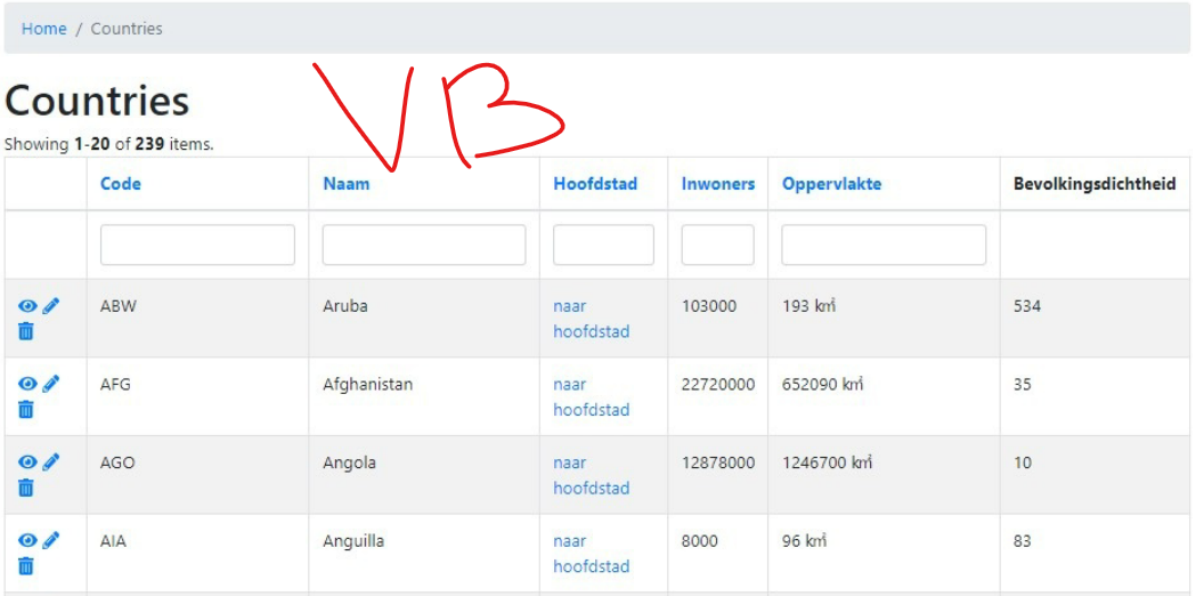
bevolkingsdichtheid = aantal inwoners / oppervlakte.

Stel een land heeft een oppervlakte van 100 km2 en je er wonen 200 mensen. Dan heb je in dit land dus gemiddeld 2 inwoners per km2, want:

```
2 = 200 / 100
```

- Maak nu een nieuwe kolom in het *country* overzicht en bereken daarin de bevolkingsdichtheid.
- Gebruik de php functie *round()* om de bevolkingsdichtheid met 0 decimalen af te drukken.

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



Het Eindresultaat

? Inleveren

1. Maak een **schermafdruk** van jouw Country's aangepaste *Gridview* en lever deze in.
2. Stuur ook jouw (aangepaste) **country view** in.

--

Revision #24
Created 2025-06-18 12:11:58 UTC by Max
Updated 2026-07-05 08:41:26 UTC by Max