

# Yii 2

## *Debug ready*

## Opdracht debugging

Om straks de opgave goed te kunnen maken, gaan we een eenvoudige debug functie maken.

### Easy Debugging

Als je de aanpassingen in dit onderdeel uitvoert, dan kun je later in de code de volgende twee functies gebruiken:

`_d($objectnaam)` bekijk \$object de code loopt gewoon door.

`_dd($objectnaam)` bekijk \$object de uitvoering van de code stopt.

Hoe krijg je dit? Eenvoudig twee stappen.

#### Stap 1 , New file config/functions.php

```
<?php
/**
 * Debug function
 * _d($var);
 */
function _d($var,$caller=null)
{
    if(!isset($caller)){
        $caller = debug_backtrace(1)[0];
    }
    echo '<code>Line: '.$caller['line'].'<br>';
    echo 'File: '.$caller['file'].'</code>';
    echo '<pre>';
    yii\helpers\VarDumper::dump($var, 10, true);
    echo '</pre>';
}
```

```
}

/**
 * Debug function with die() after
 * _dd($var);
 */
function _dd($var)
{
    $caller = debug_backtrace(1)[0];
    _d($var,$caller);
    die();
}
```

**Stap 2** , Edit config/web.php - voeg regel 3 en 4 toe.

```
<?php

/* Include debug functions */
require_once(__DIR__.'/functions.php');
```

## Inleveren

Jouw aangepaste functions.php

--

## *Version control, GIT en VSC*

Als je met een framework werkt, pas je vaak meerdere bestanden aan. Soms maak je fouten en weet je even niet meer waarom het opeens niet meer werkt. Dan is het handig als je even naar de vorige stap terug kan.

Dat kan als je *version control* toepast. In VCS kun je GIT daarvoor gebruiken.

In de module DevOps is uitgelegd hoe dit werkt. Als deze module niet beschikbaar is, dan loop je wat voor, en zul je zelf even moeten opzoeken hoe je GIT en VSC kan gebruiken.

Als GIT goed is geïnstalleerd, dan zie je het volgende in jouw VSC-editor.

Screenshot 2023-02-13 171251.png

Bij het rode rondje zie je de wijzigingen die je hebt gemaakt.

Bij het oranje rondje kun je een korte versienaam intypen en met commit, kun je deze versie dan 'committer'.

## *Inleveren*

Laat zien dat GIT werkt in je browser door een scherm afdruk van je gehele VSC in te leveren.

## *Yii Refresh*

## *Grid view (2)*

Zorg ervoor dat jouw Yii World project uit Yii Level 1 weer (nog) draait

In deze opdracht hebben we een refresh en gaan we nog een keer kijken naar de standaard Yii gridview.

Wat leren we?

- We frissen onze kennis weer wat op en leren met het MVC model te werken.
- Hoe we de standaard Yii grid view kunnen tweaken (aanpassen).
- Hoe je in PHP een af kan ronden.
- Hoe je de vormgeving met CSS aan kunt passen.

## Checklist

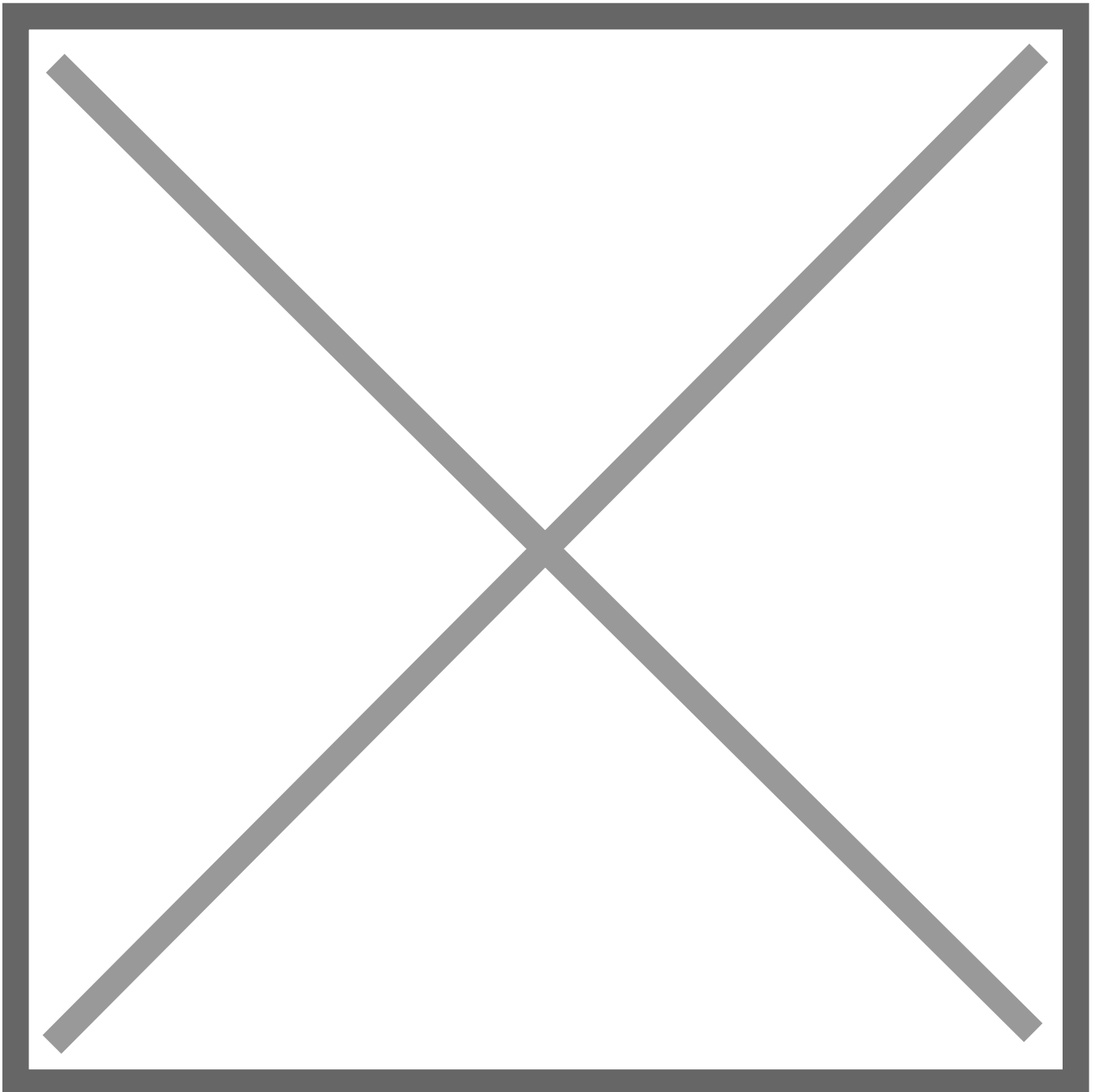
1. Controleer of je de views Country, City en CountryLanguage nog hebt. Heb je de modellen en de controllers?

2. Open jouw Yii project in VSC
3. Controleer de map models, heb je de 6 bestanden: City.php, CitySearch.php, Country.php, CountrySearch.php, CountryLanguage.php en CountryLanguageSearch.php?  
Als je deze bestanden niet hebt, ga dan terug naar Yii deel 1 en doorloop nog een keer alle stappen.
4. Type `php yii serve` in het terminal window in VCS in en controleer op localhost:8080/country of you website het nog doet.

## OK all fine, let's go!

Om er weer in te komen gaan we onze country view aanpassen.

Dit moet er uit komen:



In stap 1 is dit wat er moet uitkomen. Controleer de kolommen. Heb je deze kolommen al? Dat kan want de laatste opgave van Yii Level 1 ging ook over dit onderwerp. Lees de opgave toch even door want je moet het overzicht precies zo maken als in het voorbeeld.

## Stap 1 - kolomnamen naar rechts uitlijnen

De laatste drie kolommen zijn naar rechts uitgelijnd.

De header kan je met de volgende code naar rechts uitlijnen.

```
'headerOptions' => [ 'style' => 'text-align:right;' ],
```

Dit voeg je toe aan het array waarin je de kolom in de GridView.

Dat is het hele code blok dat tussen [ en [ staat.

## Stap 2 - getallen naar rechts uitlijnen

Zorg ervoor dat de inhoud van de kolommen ook naar rechts uitlijnt. Voeg hiervoor de juiste CSS toe aan het codeblokje dat over de kolom gaat.

Weet je niet meer hoe het werkt, kijk dan nog een keer hoe je dit in Yii Level hebt gedaan.

## Stap 3 - bevolkingsdichtheid 2 decimalen

Bevolkingsdichtheid rond je af op 2 decimalen (is nu 0).

## Stap 4 - inwoneraantal met " *thousand seperator* "

In de kolom Inwoners formatteer je het getal zodat er een spatie tussen de getallen komt. Dit is de zogenaamde "*thousand seperator*". Gebruik hiervoor de PHP functie "*number\_format*".

## Stap 5 - 0 inwoners wordt onbewoond

Landen als *Antarctica* hebben 0 inwoners. Vervang de 0 door " *Onbewoond* ".

Tip: met "if ( \$data->Population == 0 ) { ...}" kan je testen of het aantal inwoners 0 is.

## Stap 6 - Kolom een kleur geven

De kleur van het font van de kolom Naam wordt donkerblauw (CSS *darkblue* ). Zie voorbeeld.

Het font blijft wel bold.

Check of alles goed werkt!

# Inleveren

1. Screenshot van je aangepaste city view zoals het voorbeeld hierboven. Laat een compleet screenshot zien zodat ook je datum en tijd rechtsonder in het beeld is te zien.
2. Jouw aangepaste code ( `view/country/index.php` ).

## Menu (oops)

# Inleiding

*We gaan een eigen overzicht maken. We beginnen met het aanmaken van een menu, dan een controller en daarna ook nog een view. We leren data op te halen vanuit de controller en tonen die vervolgens onze eigen view.*

*Het ophalen van data doen we met een eenvoudige query die we in de Controller plaatsen. Vanuit de Controller gaan we via het Model naar de database. De volgorde wordt:*

# Menu

Om te beginnen voegen we een nieuwe methode (=functie) toe aan de **controllers/CountryController** class. De methode geeft voor nu alleen "oops" terug en doet verder nog even niets. We noemen de methode `actionOverzicht`, door de hem zo te noemen weet Yii dat het deze functie aan moet roepen als de gebruiker de url **`/country/overzicht`** aanroept. Het koppelen van een url aan een stuk logica, zoals een functie of methode noemen we **routing** .

# Opdracht

Maak een menu item `overzicht` onder `country` dat naar `/country/overzicht` gaat.

In [Yii level 1](#) hebben we met menu's geoeft.

[image-1614620329168.png](#)

Test het menu uit.

Het menu gaat naar de actionOverzicht in de CountryController en returned "oops". We krijgen dus een leeg scherm met "oops".

## Inleveren

- Een schermafdruck van jouw gehele browser (url moet zichtbaar zijn) waarin je het via het menu naar jouw overzicht gaat en waar "oops" wordt afgedrukt.

# Controller

*We gaan in onze nieuwe controller functie actionOverzicht code plaatsen, deze code haalt alle data van alle countries op en roept vervolgens de view overzicht aan.*

## Controller

We halen onze "oops" weg en gaan dus een stukje code toevoegen aan onze CountryController.

Let op zet jouw naam op regel 3.

```
public function actionOverzicht()
{
    // Zet hier jouw naam.
    // Dit is de query, dit is te vergelijken met select * from Country
    $countries = Country::find()->all();
```



```
// De code hieronder genereert de html, dit noemt men de view binnen MVC (model view controller).  
// Zoals je kunt zien geven we de $countries array door aan de "overzicht" view.  
// In de overzicht View kunnen we nu $countries gebruiken.  
  
return $this->render('overzicht', [  
    'countries' => $countries,  
]);  
}
```

## Opdracht

Maak de function `actionOverzicht` in de `CountryController` zoals hierboven is aangegeven.

Als je alle stappen goed hebt doorlopen krijg je als deze code aanroept een foutmelding.

image-1614623472969.png What? Een fout? Waarom nu?

Leg uit waarom je een foutmelding krijgt.

## Inleveren

1. Uitleg waarom je een foutmelding krijgt.  
(maak een .txt bestand met notepad en zet daarin de uitleg)
2. CountryController.php

## View

## View

Het menu 'overzicht' gaat naar de `actionOverzicht` deze voert een query uit en roept de view overzicht op. Dit overzicht gaan we nu maken.

We gaan dus in de folder `views/country` een nieuwe file `overzicht.php` aanmaken en plaatsen daar de volgende code in.

```
<?php
    foreach ($countries as $country) {
        echo $country->Name;
        echo " - ";
        echo $country->Code;
        echo " - ";
        echo number_format($country->Population, 0, ',', ' ');
        echo "<br>";
    }
?>
```

## Opdracht

Maak de view `overzicht` zoals hierboven is aangegeven.

Het overzicht is bijna niet geformatteerd.

[image-1614627165703.png](#)

Weet je nog hoe een HTML table er uit ziet?

```
<table>

<tr>
  <td> .. </td>
  <td> .. </td>
  <td> .. </td>
</tr>

<tr>
  <td> .. </td>
  <td> .. </td>
  <td> .. </td>
</tr>

</table>
```

Hierboven staat een skelet van een table met twee rijen en drie kolommen.

Vind je dit lastig? [Kijk dan de video.](#)

Verander de `/view/country/overzicht` nu zo dat het netjes in een table wordt gezet. De output komt er als volgt uit te zien:

[image-1614627534845.png](#)

De kolommen staan nu dus netjes onder elkaar.

Je hebt nu zelf een view opgebouwd zonder gebruik te maken van de Gridview widget. Het kost meer tijd en moeite om zelf een overzicht zonder Gridview widget te maken, maar je hebt wel veel meer controle over hoe jouw overzicht er uit komt te zien.

## Inleveren

- Een screenshot van je browser (met URL) met daarin het overzicht dat je hebt gemaakt en
- de aangepaste en werkende `view/country/overzicht.php`

--

## Europa

We hebben een eigen overzicht gemaakt. Dat gaan we nog een keer doen en nu met alle landen van Europa, een selectie dus.

## Opdracht

- Maar een nieuw menu-item en noem dit " *Overzicht Europe* " .
- Maak een nieuwe function `actionOverzichtEurope`, zet deze in de juiste controller en koppel deze aan het menu-item.

(let op de vertaling naar de routing; in de routing staat nooit een hoofdletter en woorden worden gescheiden door een streepje "-", [zie les 2 over routing](#) ).

- Laat in dit overzicht alle landen van Europa zien. Op de vorige pagina staat uitgelegd hoe dit moet.
- Druk de volgende kolommen af:  
*Name* en *Surface Area*
- Zet de twee kolommen netjes onder elkaar (in table)
- Formateer de Surface Area netjes met spatie tussen de duizendtallen en lijn ze recht uit.
- Maak de headers **bold** .
- Sorteert het overzicht op Surface Area van grootste land naar kleinste.

Het overzicht komt er dus zo uit te zien:

[image-1614629726011.png](#)

Je hebt geleerd dat je een overzicht kan maken via een (Yii-)query. Met de query kan je de selectie en sortering aanpassingen. De query wordt geplaatst in de *controller* en maakt gebruik van het model. In dit voorbeeld het model van *country* .

## Inleveren

- Jouw aangepaste en werkende view/country/overzicht.php in tabel vorm, en
- Een schermafbeeld yii-04a-jouw-naam van jouw gehele browser waarin je je overzicht laat zien.

***Selectie (SurfaceArea < 100000)***

## Opdracht

- Pas het overzicht van de vorige opdracht aan zodat je alleen alle kleine Europese landen ziet.

- Een land is klein als

```
SurfaceArea  
<  
100000
```

Pas hiervoor de controller aan.

- Je hebt dus te maken met **twee** zoekcriteria.

Zoek zelf op internet uit hoe je in een 'Yii-query' twee zoekcriteria kan combineren met *andWhere()* of gebruik de query methode om met Yii data uit de database te halen.

## Inleveren

- Jouw aangepaste en werkende countryController.php

## Hoofdstad

## Opdracht

**Lees** de uitleg (op de vorige pagina) goed door en plaats de kolom hoofdstad in het overzicht.

[image-1615887892142.png](#)

Maak hiervoor twee aanpassingen:

1. maak de relatie in het model van *Country* , en
2. pas de view *overzicht* aan door daar de kolom *Hoofdstad* aan toe te voegen.

# Inleveren

1. Een schermafbeeld yii-05a-<jouw naam> van je country view waarin de kolommen Naam, Hoofdstad en Oppervlakte worden afgedrukt, zoals in het voorbeeld hierboven. Maak een afdruk van je gehele browserscherm.

## *Clickable hoofdstad*

We gaan de Hoofdstad clickable maken. Als je naar <http://localhost:8080/city> gaat dan zie je het overzicht van steden. Klik op een oogje op een van de steden. Let op de URL. Je ziet <http://localhost:8080/city/view?id=<NR>> als url en <NR> is het ID van de stad.

Dus als je bijvoorbeeld klikt op <http://localhost:8080/city/view?id=5> dan zie je informatie over Amsterdam.

## Opdracht

Maak elke hoofdstad in het overzicht dat je bij opdracht 1 hebt gemaakt *clickable*. Als je op de link *klikt dan* open je het overzicht van de stad, bijvoorbeeld <http://localhost:8080/city/view?id=5> .

Tip: je kunt `Html:a()` gebruiken om een link te maken. Dit is in de [vorige les uitgelegd](#) .

[image-1615896595590.png](#)

# Inleveren

1. Jouw aangepaste country view. Plaats in je country view commentaar met jouw naam.

## *Eén taal*

# Opdracht

**Lees** de uitleg hierboven goed door. Druk in de laatste kolom n taal af.

[image-1616504705123.png](#)

Nu hebben we n taal per land, maar in de meeste landen wordt meer dan n taal gesproken. Daar gaan we in de volgende opdracht naar kijken.

Let op: je hebt in je model een query staan die alleen alle landen uit Europa selecteert. Zorg dat je hierbij alle kolommen selecteert, omdat je anders de kolommen mist die je nodig hebt om de link (join) te maken.

Dus dit is onjuist:  
image.png

*In dit voorbeeld selecteer je maar drie kolommen (Name, SurfaceArea en Capital) en de kolom Code waarmee je de join naar de tabel Countrylanguages moet maken zit er niet bij. Je voegt dus de juiste kolommen toe of je laat de hele select regel weg zodat alle kolommen worden geselecteerd.*

## Inleveren

Een schermafbeelding yii-06a-jouw-naam met het scherm van jouw gehele browser waarin je het landenoverzicht laat zien en waarbij je n taal afdruckt.

***Meer talen***

Om alle talen die bij n land horen af te drukken, gaan we met een loop door het array \$country->countrylanguages heen.

Gebruik hiervoor de volgende loop.

```
foreach( $country->countrylanguages as $taal) {  
    // vul zelf de juiste variable in om de taal af te drukken  
    echo .....;  
    echo " <br/>";  
}
```

## Opdracht

Gebruik de loop zoals hierboven is aangegeven, vul deze aan en maak het volgende overzicht.

[image-1616505080100.png](#)

Zorg ervoor (met CSS) dat de tabel er netjes uitgelijnd uit ziet zoals hierboven in het voorbeeld.

Je hebt nu een 1:N ( n op meer) relatie gemaakt. In volgende lessen gaan we hier opnieuw mee aan de slag.

## Inleveren

- Een schermafdruck yii-06b-jouw-naam.png waarin je je complete scherm laat zien waarin het landenoverzicht te zien is met alle talen zoals in het voorbeeld.

# *Percentage*

## Opdracht



In de Countrylanguages tabel staat naast de taal ook het percentage. Dit is het percentage van de bevolking dat deze taal spreekt. Zet dit percentage tussen haakjes achter elke taal. Voorbeeld:

[image-1616505413868.png](#)

## Inleveren

- Een schermafdruck yii-06c-jouw-naam.png waarin je je complete scherm laat zien waarin het landenoverzicht te zien is met talen en percentages zoals in het voorbeeld.

## Voorbeeld

Screenshot 2022-12-25 101520.png

## *Sorteren*

## Opdracht

Sorteer de talen op percentage aflopend van hoog naar laag. Op die manier komt de meest gesproken van een land bovenaan.

[image-1616506111435.png](#)

Tip: gebruik Google om te zoeken hoe je de relatie in het model verandert zodat je **aflopend** kunt sorteren.

## Inleveren

1. Een schermafdruck yii-06d-jouw-naam.png waarin je je complete scherm laat zien waarin het landenoverzicht te zien is zoals in het voorbeeld.
  2. De code van de view van country. Voeg in jouw code commentaar toe waarin je jouw naam zet.
-

Revision #2

Created 2024-05-30 17:45:43 UTC by Max

Updated 2024-10-16 06:55:13 UTC by Max