

5, Menu - Controller - View

We gaan een eigen overzicht maken. We beginnen met het aanmaken van een menu, dan maken we een controller, en van de controller gaan we naar de view. We leren data op te halen met de controller en deze te tonen in 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:

Gebruiker -> (kiest) menu -> controller -> model -> database -> (resultaat terug naar) controller -> view -> gebruiker.

In deze les maken we dus het hele 'rondje'. Dit is een standaard 'rondje' dat in elke applicatie vaak terug zal komen.

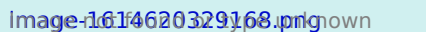
Menu - Controller - View

Om te beginnen maken we een nieuwe method (=functie) in de **controllers/CountryController** class.

Deze controller noemen we actionOverzicht. Volgens de routing regels wordt de route wordt dus /country/overzicht.

Opdracht 1

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



Probeer het menu uit. Wat zie je en kun je verklaren wat je ziet?

Juist.... je krijgt een foutmelding want de routing verwijst naar de method/function actionOverzicht in de controller CountryController en die bestaat niet!

We gaan dus een stukje code toevoegen aan onze CountryController.

```
use yii\data\Pagination;
```

```
..
```

```
..
```

```

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

    // de view wordt aangeroepen en het object $countries en $pagination wordt meegegeven.
    return $this->render('overzicht', [
        'countries' => $countries,
    ]);
}

```

Opdracht 2

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

Test opnieuw wat er gebeurt als je het menu *overzicht* selecteert

image-1614623472969.png

What? Weer een fout? Waarom nu?

Juist het menu gaat naar de *actionOverzicht* en de *actionOverzicht* voert een query uit en roept de view *overzicht* op. En die view bestaat (nog) niet.

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 3

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>

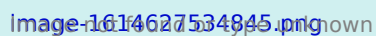
</table>
```

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

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

Opdracht 4

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.

Sorteren en selecteren

Via de Yii Gridview widget hebben we geleerd dat we eenvoudig kunnen sorteren en selecteren. De Yii Gridview heeft ook beperkingen. Je kunt bijvoorbeeld niet alle landen zoeken met *meer dan* X

inwoners. Soms wil je ook dat de gebruiker zelf niet kan selecteren. Hij krijgt dan niet alle regels uit de tabel te zien, maar bijvoorbeeld alleen de regels met de datum van vandaag.

We gaan het sorteren en selecteren via de controller coderen.

Kijk nog eens naar de *CountryController* bij de *actionOverzicht*, daar staat:

```
$countries=Country::find()->all();
```

Dit statement kijkt naar het object *Country* dat in het model is gemaakt en zoekt dan naar alle regels uit dit object. Dit object wordt in het model gekoppeld aan de tabel *country*.

Yii vertaald dit naar de query

```
SELECT * FROM country
```

We kunnen nog veel meer, kijk maar eens naar het volgende uitgebreide voorbeeld:

```
$countries = country::find()->where(['Continent' => 'Europe']->orderBy(['Name' => SORT_ASC])->all();
```

Hier worden alle *countries* geselecteerd met *Continent=Europe*, gesorteerd op *Name*.

Dit is een soort Yii query; een query in Yii-formaat.

Het SQL-statement zou er zo uit zien:

```
SELECT * FROM country WHERE Continent = 'Europe' ORDER BY Name ASC
```

Op deze manier kun je dus via de code in de controller een selectie maken en de sortering aanpassen. In de volgende opdracht komt dit terug.

Opdracht 5

- Maak 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.

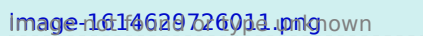
(let op de vertaling naar de routing; in de routing staat nooit een hoofdletter en woorden worden gescheiden door een streepje "-", [zie les 2 over routing](#)).

- Laat in dit overzicht alle landen van Europa zien. Gebruik hiervoor het stukje code dat hierboven staat (`$countries = country::find()->where.....`).
- Druk de volgende kolommen af:

Name en Surface Area

- Zet de twee kolommen netjes onder elkaar
- Formateer de Surface Area netjes met spatie tussen de duizendtallen en lijn ze recht uit.
- Maak een header die **bold** is.
- Sorteer het overzicht op Surface Area van grootste land naar kleinste.

Het overzicht komt er dus zo uit te zien:

The image is a placeholder for a screenshot of a web application. It shows a table with two columns: 'Name' and 'Surface Area'. The 'Name' column contains the names of European countries, and the 'Surface Area' column contains their surface areas, formatted with spaces as thousands separators and aligned to the right. The table has a header row with bold text. The image is labeled '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 aanpassen. De query wordt geplaatst in de *controller* en maakt gebruik van het model. In dit voorbeeld het model van *country*.

Dit komt terug in het examen, daarom nog een opgave over dit onderwerp.

Opdracht 6

Pas het overzicht van Opdracht 5 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()*.

--

Revision #56

Created 9 July 2020 21:12:14 by Max

Updated 21 June 2021 09:30:50