

Yii 3

Koffie opdracht

De opdracht

Maak de *coffee* database met de drie tabellen, *menu* , *medewerker* en *bestelling* .
Maak voor elke tabel een CRUD.

Volg de stappen hieronder om dit uit te voeren.

In stap 1 maak je een nieuwe database

In stap 2-6 maak je een nieuw Yii project aan. De uitgebreide instructie met voorbeelden kan je vinden in les 1.

In stap 7-12 maken we de tabellen en de CRUD pagina's.

Voer stap 1 tot en met 13 uit.

Stap 1, maak een database *coffee*.

Stap 2, maak een nieuw Yii project *coffee*. Gebruik de onderstaande regel voor in de Terminal.

```
CREATE DATABASE coffee;
```

Let op dat je `<naam van de map hier!!>` vervangt voor de naam van dit project

```
composer create-project --prefer-dist yiisoft/yii2-app-basic <naam van de map hier!!>
```

Stap 3, pas de database configuratie file in de Yii aan en laat deze verwijzen naar de *coffee* database.

Stap 4, pas de web.php in de Yii config aan en [zet smart routing aan](#) (uncomment het gedeelte bij 'urlManager').

Stap 5, maak de file functions.php met daarin de functies dd() en d(). Dit is beschreven in [les 1](#) .

Stap 6, roep functions.php aan vanuit de web.php configuratie file. Ook dit staat in les 1 beschreven.

Stap 7, we gaan nu de tabellen, models en CRUD's maken voor medewerker, menu en bestelling. Hieronder vind je terug hoe je de tabellen moet maken.

medewerker

Maak de tabel in de database.

Gebruik hiervoor localhost/phpmyadmin (hiervoor moet XAMPP draaien)

[image-1616594908052.png](#)

Image not found or type unknown

Genereer het **model** en de **CRUD** voor *medewerkers* in Yii en zet de volgende namen in de database.utmelding meer krijgt.

image-1616594707813.png

Image not found or type unknown

Houd rekening met [Yii-fout die we eerder tegenkwamen](#) . Pas de gegeneerde code zodat je geen foutmelding meer ziet!

Menu

Maak nu de tabel menu in de database

image-1616595011664.png

Image not found or type unknown

Stap 10, Genereer het model en de CRUD voor *menu* en zet de volgende namen en prijzen in de database.

De prijzen zetten we in de database als centen. Dus E 1.95 wordt 195. Dit maakt het coderen eenvoudiger omdat we op deze manier met integers (gehele getallen kunnen werken).

Zet de testdata in de database zoals je hieronder kan zien.

image-1616594739223.png

Image not found or type unknown

Bestelling

Maak nu de tabel bestelling in de database.

image-1616595477669.png

Image not found or type unknown

Stap 12, Genereer het model en de CRUD voor *bestelling*. (er hoeft geen data in de database gezet te worden voor *Bestelling*)

We hebben nu een tweede standaard Yii applicatie en we gaan nu aanpassingen maken in het invoerformulier 'Create Bestelling'.

Als we nu een nieuwe bestelling maken dan zien we het volgende scherm:

Image not found or type unknown

Stap 13 , *timestamp* halen we weg, want die wordt automatisch ingevoerd (default). Die hoeven we dus **niet** in te voeren.

Open de *_form.php* in de view van bestelling en haal de *timestamp* regel er uit.

Controleer in alle drie de CRUD's of ID in het invoerveld staat. Controleer hiervoor de drie *_form.php* bestanden in de views van medewerker, bestelling en menu.

Haal de ID's uit het invoer forms en test of je gegevens kan invoeren.

Inleveren

1. Schermafdruk van de browser met je *form.php* van de view *bestelling*. Noem het bestand *yii-07-jouwnaam.png*
2. het bestand *_form.php* van de view bestelling (dus *views/bestelling_form.php*). Zet jouw naam in commentaar in dit bestand.

Schermafdruck ziet er zo ongeveer uit (je moet wel je hele browserscherm inleveren!).
Screenshot 2022-11-30 160819.png

Image not found or type unknown

--

Forms en Dropdown

Opdracht

Maak nu een statische drop down met de waarden zoals in het voorbeeld van de vorige pagina (Ayoub, Brahim, Carla,).

Statisch betekent dat de waarden niet veranderen. Het tegenovergestelde is dynamische en dat betekent dat de waarden veranderen, omdat deze waarden bijvoorbeeld uit de database komen.

De waarden worden dus (nog) niet uit de database gehaald.

Inleveren

1. Schermafdruck yii-08a-jouw-naam met het form waarin je de drop down (opengeklapt) laat zien. Maak een schermafdruck van je **gehele** browser.

Voorbeeld

Image not found or type unknown

Dynamische drop down

Opdracht, Dynamische drop down

In de vorige opdracht hebben we een drop down van medewerkers gemaakt, maar we willen de lijst van medewerkers natuurlijk uit de database halen.

In deze opdracht leg ik stap-voor-stap uit wat je moet doen om dat voor elkaar te krijgen.

Lees alles aandachtig door en sla geen stappen over!

Data uit de database halen doen we in de controller.

Open de `BestellingController`. Vanuit deze code wordt de create view aangeroepen en vanuit de create view wordt de `_form.php` aangeroepen. Waarom dit in twee stappen gaat leggen we later uit.

We veranderen de code in `Bestelling controller`:

```
use app\models\Medewerker;
use Yii;
..
..

public function actionCreate()
{
    $model = new Bestelling();
    $medewerkers = Medewerker::find()->all();

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render('create', [
        'model' => $model,
        'medewerkers' => $medewerkers
    ]);
}
```

De eerste regel `use app\models\Medewerker;` zetten we bovenaan in de `BestellingController`. Hiermee vertellen we Yii dat vanuit de `BestellingController` gebruik willen gaan maken van het model `Medewerker`.

Let op dat je op de bovenaan in je code `use Yii;` moet hebben staan.

Let ook op wat je moet aanpassen, de code die hierboven staat kan in een nieuwere Yii versie iets anders zijn. Kopieer **alleen** de regels die er voor zorgen dat de create view de `$medewerkers` krijgt.

We kunnen de medewerkers ook ophalen via een sql-query. Je **mag** regel 10 ook vervangen in de volgende twee regels:

```
// $medewerkers = Medewerker::find()->all();
// dit is hetzelfde als
```

```
$sql="select * from medewerker";  
$medewerkers = Yii::$app->db->createCommand($sql)->queryAll();
```

Vanuit de controller zie je dat de `$medewerkers` (waar dus alle medewerkers in staat) wordt doorgegeven aan de `create` view (van bestelling).

Dus wat hebben we gedaan tot nu toe?

In de function `actionCreate` die wordt aangeroepen als we een nieuwe bestelling willen maken, voegen we een object toe waarin alle medewerkers zitten. In regel 10 halen we alle medewerkers op en in regel 18 sturen we het resultaat naar de view `create.php` (van `bestelling`).

In de view `create.php` van bestelling controller passen we het laatste stukje code aan

```
<?= $this->render('_form', [  
    'model' => $model,  
    'medewerkers' => $medewerkers  
) ?>
```

Hiermee geven we het object `$medewerkers` door aan de view `_form` van bestelling.

We hebben dus deze volgorde afgelegd: `bestellingController.php` (controller) -> `create.php` (view) -> `_form.php` (view)

In het view `_form` zetten we nu bovenaan in het PHP-gedeelte.

```
_dd($medewerkers);
```

Dit is de debugfunctie en hiermee controleren we of we inderdaad alle medewerkers naar de view hebben gestuurd.

We hebben nu als het goed is een lijst van medewerkers in de `_form` maar we moeten het ombouwen naar een associatieve array.

Pas hiervoor de code aan in de view `_form` van bestelling.

```
<?php  
  
use yii\helpers\Html;  
use yii\widgets\ActiveForm;  
use yii\helpers\ArrayHelper;
```

```
// hiermee maken we een array met de id's(als key) en de name (als value)
$medewerkerList = ArrayHelper::map($medewerkers,'id','naam');
_dd($medewerkerList);
?>

...

...

...
```

Deze code laat het eerst stukje van de view_form zien.

Met de functie ArrayHelper::map zetten we het object \$medewerkers om in een associative array. Met _dd() laten we dat zien. Probeer maar! Als het goed is, zie je het volgende:

```
[
    1 => 'Ayoub'
    2 => 'Brahim'
    3 => 'Carla'
    4 => 'Diego'
    5 => 'Eisa'
]
```

En dit is precies wat we nodig hebben om de Drop Down te maken.

Pas de regel in de view_form van bestelling waarin de user het id van de medewerker moet intypen aan. Verander deze regel in:

```
<?= $form->field($model, 'medewerker_id')->dropDownList($medewerkerList, ['prompt' => ''])-
>label('Medwerker') ?>
```

Als het goed is heb je hiermee een werkend menu gekregen.

Image not found or type unknown

Gelukt? Wordt de lijst van medewerkers in de drop down uit de database gehaald?

Inleveren

1. de view _form.php van bestelling
2. schermafdruck yii-08b-jouw-naam met het form waarin je de drop down van de medewerkers (opengeklapt) laat zien. Maak een schermafdruck van je gehele browser

--

Nog een drop down

We gaan nog een keer stap-voor-stap alle stappen uitvoeren om nog een drop down in bestelling te maken.

Opdracht

Maak nu een drop down voor de bestelling zodat de medewerker uit een lijstje van koffiesoorten kan kiezen bij het opnemen van de bestelling.

[image-1616605448841.png](#)

Image not found or type unknown
Gebruik hiervoor het stappenplan dat hieronder is beschreven.

Het stappenplan voor een drop down

Schrijf eerst het volgende op:

In welk model komt de drop down? Dit noemen we de target.
(in opdracht 2 hierboven is dat *Bestelling*)

Waar komt de informatie voor de Drop Down uit? Dit noemen we de source.
(in opdracht 2 hierboven is dat *Menu*)

Voer nu de stappen 1-6 uit

1. Zet in de target controller welke source model we willen gebruiken.

```
// in controller van target  
use app\models\<SourceModelnaam>;
```

2. Haal in de target controller in de actionCreate() de informatie uit de source op.

```
// sourceName bedenkt u zelf en daarin staat de informatie die in de drop down moet komen.  
$sourceName = source::find()->all();
```

3. Geef vanuit de target controller in de actionCreate() de informatie door aan de view.

```
return $this->render('create', [
    'sourceName' => $sourceName,
    ....
]);
```

4. Open de view create in de target (view) en geef de informatie door aan `_form` view.

```
<?= $this->render('_form', [
    'sourceName' => $sourceName,
    ...
]);
```

5. Open de view `_form` in de target (view) en zet de informatie om in een associative array

```
// in dit voorbeeld worden id en naam gebruikt:
// id is het veld dat moet worden opgeslagen
// naam is het veld dat in de drop te zien is
$sourceNameList = ArrayHelper::map($sourceName, 'id', 'naam');
```

6. Pas het invoerveld aan in de `_form` van de target (view).

```
// veld_in_de_db is het veld van de target dat in dit invoerveld wordt weggeschreven.
<?= $form->field($model, 'veld_in_de_db')->dropDownList($$sourceNameList, ['prompt' => '']) ?>
```

Als je alles correct gedaan hebt dan moet je nog 1 regel aanpassen in de code om de functionaliteit te gebruiken. Yii controleert automatisch of alle databasekolommen zijn ingevuld in het formulier. Dit controleren wordt *validatie* genoemd. Deze regels staan in het model. Omdat je geen Bestelling ID opgeeft bij het maken van een bestelling moet je deze requirement verwijderen op regel 33 van model Bestelling.

```
return [
    ['naam', 'medewerker_id', 'menu_id', 'status'], 'required'],
    ...
    ...
]
```

Inleveren

1. schermafdruck yii-09-jouw-naam met het form waarin je de drop down van de producten (opengeklapt) laat zien. Maak een schermafdruck van je gehele browser.

2. de *BestellingController.php* en zet jouw naam bij de aanpassingen die je hebt gedaan.

Drop down afmaken

Opdracht

Zorg ervoor dat de update weer werkt en dat de objecten op de juiste manier worden doorgegeven. Lees hiervoor de uitleg die op de vorige pagina staat.

Pas de `actionUpdate` in de `BestellingController` aan.
Pas de update view van de `BestellingController` aan.

Zorg dat de update van een `Bestelling` goed werkt.

Inleveren

1. de *BestellingController.php* en zet jouw naam bij de aanpassingen die je hebt gedaan.

Final Touch

Opdracht

De final touch

Laten we nog wat labels aanpassen. Dit zijn de teksten die boven de invoervelden in het form staan.

Pas de labels in het form aan zodat er Medewerker, Klantnaam, Bestelling en Status Bestelling komt te staan:

[image-1616611684520.png](#)

Image not found or type unknown

Zoek zelf uit hoe dit moet. Tip gebruik de zoektermen:

How to change label text of the ActiveForm?

Inleveren

1. schermafdruck yii-010b-jouw-naam met het bestellingen form waarin je de aangepaste labels laat zien. Maak een schermafdruck van je gehele browser.

Relaties en drop down

Opdracht

Zorg er nu voor dat je vanuit de controller de medewerkers opvraagt en deze verstuurd naar de index view.

Gebruik de

`ArrayHelper::map()`

functie om je object om te zetten in een list.

En gebruik de list dan in de gridview.

In de kolom medewerker zien we nu de id's van de medewerker. Als we de naam willen afdrukken moeten we de relatie maken. Dit hebben we eerder gedaan, zet in het model van Bestelling de volgende code:

```
public function getMedewerkers()  
{  
    return $this->hasOne(Medewerker::className(), ['id' => 'medewerker_id']);  
}
```

Plaats nu in de gridview in de index view van de bestelling:

```
[  
    'attribute' => 'medewerker_id',  
    'label'      => 'Medewerker',  
    'filter'     => $medewerkerList,  
    'value'      => 'medewerkers.naam'  
],
```

Hiermee zetten we de waarde van deze kolom op *naam* van de relatie *medewerkers* .

Inleveren

1. schermafdruck yii-011a-jouw-naam met de bestellingen index view waarin je laat zien dat de relatie is gemaakt met de tabel medewerker. Maak een schermafdruck van je **gehele** browser.

Medewerker tonen

Opdracht

Pas tenslotte het label aan en zorg ervoor dat de kolom waarin de medewerker wordt getoond in de gridview (van de index view van bestelling) er als volgt uit ziet:

[image-1616622153704.png](#)

Image not found or type unknown

Als er een naam wordt geselecteerd dan worden alle bestellingen die door deze medewerker zijn uitgevoerd geselecteerd.

Hiermee zetten we de waarde van deze kolom op *naam* van de relatie *medewerkers* .

Inleveren

1. schermafdruck yii-011b-jouw-naam met de bestellingen index view waarin je laat zien dat de drop down is gevuld met de namen van de medewerkers. Maak een schermafdruck van je gehele browser.

Menus en drop down

Opdracht

In deze opdracht gaan we alle stappen die we hebben uitgevoerd om de kolom *Medewerker* aan te passen nog een keer uitvoeren maar nu voor de kolom *menu_id* .

Pas de *menu_id* kolom aan in het bestellingenoverzicht (*index* view van *Bestelling*).

Zorg ervoor dat je met een drop down alle koffie soorten (*menu*) kunt selecteren. Je kunt dan dus alle bestellingen 'Americano' opzoeken.

[image-1616622602808.png](#)

Image not found or type unknown

Zorg ervoor dat je de juiste relatie legt tussen de *Bestelling* en *Menu* .

Pas dan de grid view aan zodat je in de bestelling kolom geen id's meer ziet, maar dat je de naam van de bestellingen ziet.

[image-1616622580235.png](#)

Image not found or type unknown

Let ook op het label. Dit is veranderd van *menu_id* naar *Bestelling* .

Succes!

Inleveren

1. Laat in de les zien dat alles werkt.

lukt dat niet maak dan een heel kort demo filmpje (10 to 30 seconden).

1. In dat filmpje laat je zien dat je een nieuwe bestelling aan kan maken en gebruik maakt van de drop downs in het form.
2. In het filmpje laat je zien dat je een bestelling kan aanpassen.
3. In het filmpje laat je zien dat de grid view werkt en dat via de twee drop downs een selectie kan maken.

Revision #2

Created 30 May 2024 17:49:27 by Max

Updated 30 May 2024 18:00:15 by Max