

08 Form en Drop down in bestelling

Inleiding

We gaan onze koffie applicatie uitbreiden.

Om het invoeren van gegevens eenvoudiger te maken gaan we drop down menu's toevoegen.

We gaan naar twee soorten drop-downs kijken; die met vaste waarden en die met dynamische waarden (uit de database).

Bij het invoeren van een bestelling gaan we allereerst de status via een drop down invoeren.

Daarna gaan we waarde voor de medewerker ook uit een drop down halen. Deze waarde moet uit de database komen.

We gaan eerst de insert aanpassen. Dus aanmaken van een nieuwe bestelling. De update doen we later.

Wat gaan we leren?

- hoe je in Yii in een formulier een drop-down kan maken.
- hoe je in Yii een query in de controller kan toevoegen en de resultaten kan doorsturen naar de view.

Drop down in Bestelling

We gaan kijken naar de bestelling tabel en CRUD.

Als we de status in de database als een enum hebben aangemaakt dan weet Yii dat de status uit beperkt aantal waarden kan hebben en als het goed is maakt Yii dan vanzelf een drop down in het form.

Klopt dat heb jij een drop down?

Indien niet, dan kun je de code aanpassen in het form aanpassen.

In de view `_form.php` van Bestellingen moet de volgende regel staan:

```
<?= $form->field($model, 'status')->dropDownList([ 'besteld' => 'Besteld', 'klaar' => 'Klaar', 'betaald' => 'Betaald', '' => '', ], ['prompt' => '']) ?>
```

Deze regel laat een drop down menu in het formulier zien. Kijk eens goed naar de parameter van `dropDownList`.

```
[ 'besteld' => 'Besteld', 'klaar' => 'Klaar', 'betaald' => 'Betaald', '' => '', ], ['prompt' => '']
```

De eerste parameter is een associative array waarin elk element bestaat uit een key en een value.

De key is de waarde die het element uit het form krijgt (de value) en deze waarde wordt door Yii in de database gezet. De key komt dus overeen met de waarde in de database. De value van het associatieve array is wat de gebruiker op het scherm ziet.

Deze waarde kan je dus veranderen. Je kunt dus bijvoorbeeld 'betaald' veranderen in 'afgerekend'. Het enige dat dan gebeurt is dat je iets ander op het scherm ziet.

In dit voorbeeld is de key en de value gelijk (de waarde op het scherm is hetzelfde als de waarde in de database).

Dus samengevat, de key is de waarde in de database en de value is wat de gebruiker op het scherm ziet.

Stel we willen in het form van bestelling de medewerker veranderen. Dan moeten we dus de *foreign key* die naar medewerker verwijst veranderen. Het juiste id van de medewerker moet in de tabel bestelling worden ingevuld.

De waarde wordt dus het id, maar dat is niet wat je de gebruiker wilt laten zien.

Dus als we voor de medewerkers een drop down willen maken dan hebben we een associative array nodig dat er zo uit ziet:

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

De keys zijn de id's die als foreign keys in de bestelling tabel staan en de namen zijn de namen van de medewerkers.

Opdracht 8a

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

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.

Opdracht 8b, Dynamische drop down

In opdracht 8a 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 omdat oor 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;

..

..

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*.

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 9 halen we alle medewerkers op en in regel 17 sturen we het resultaat naar de view *create.php* (van *bestelling*).

We kunnen de medewerkers ook ophalen via een sql-query. Je mag regel 9 ook vervangen in de volgende twee regels:

```

// dit is hetzelfde als
// $medewerkers = Medewerker::find()->all();
$sql="select * from medewerker";
$medewerkers = Yii::$app->db->createCommand($sql)->queryAll();

```

In de view *create.php* van *bestelling* passen we het laatste code aan

```

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

```

Hiermee geven we het object medewerkers weer door aan de view *_form* van *bestelling*.

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;

$medewerkerList = ArrayHelper::map($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_1616605065315.png](#)

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

Inleveren

1. 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.

--

Revision #2

Created 29 June 2022 12:09:12 by Max

Updated 16 July 2022 14:26:31 by Max