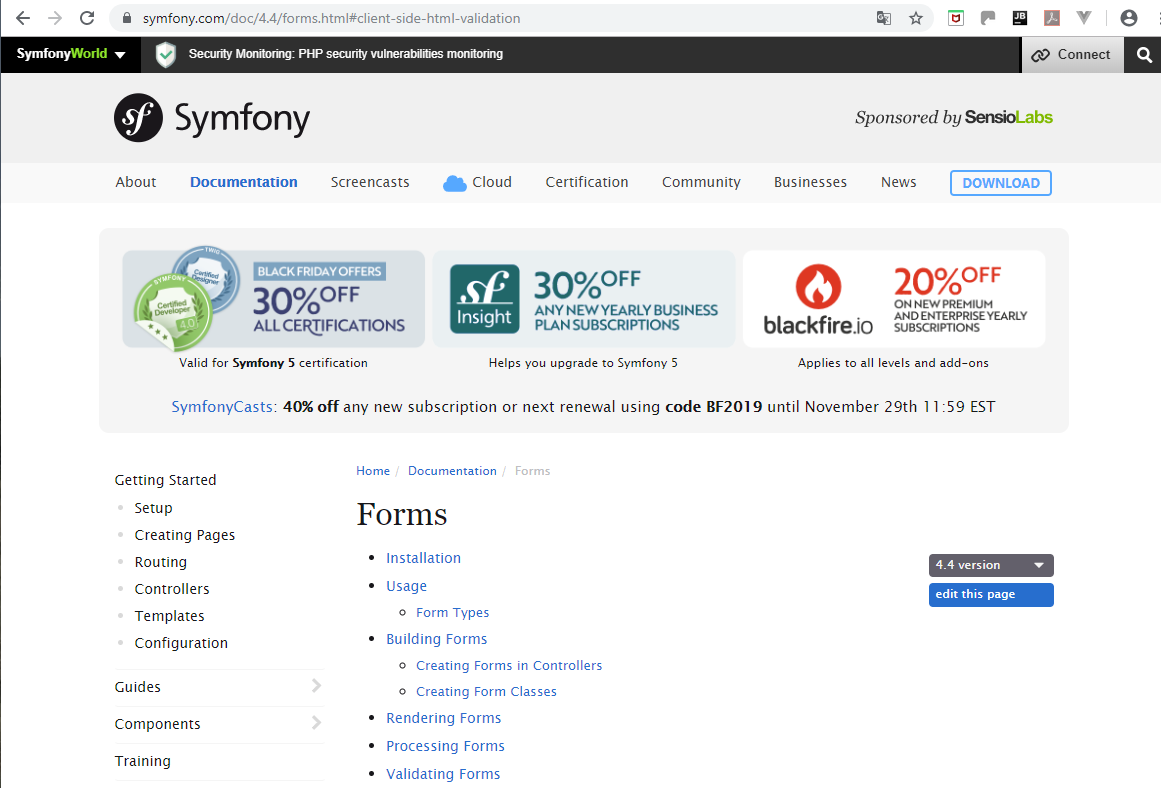
Symfony en Formulieren

Symfony en Formulieren

Ga naar de Symfony website, onderdeel Forms



1. Zorg voor een applicatie met een users- en een afdelingstabel. Deze is aanwezig na het afronden van Databases.

Voor de volgende ERD gaan we formulieren maken.

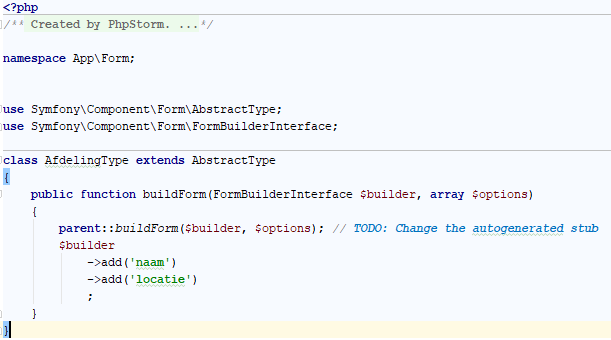


In deze tutorial gaan we een formulier maken voor de afdelings tabel.



Het maken van een formulier

Als je een formulier gaat maken moet je eerst een class maken die beschrijft hoe het formulier eruit ziet.



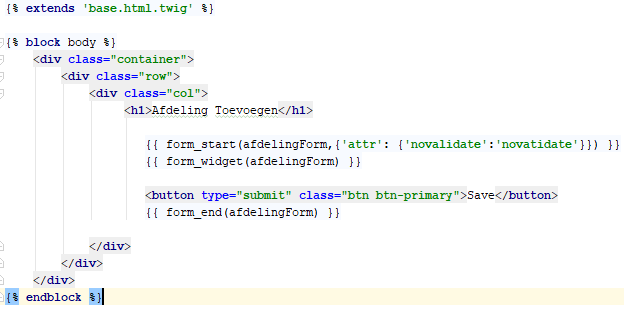
De volgende stap is dit formulier gaan gebruiken, hiervoor maken we een action aan in AdminController



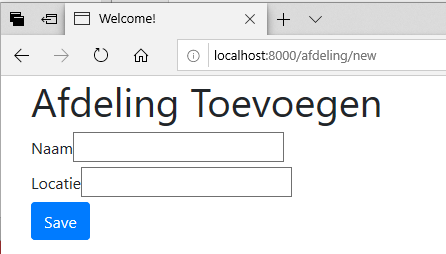
Eerst maak je een formulier object, het soort formulier geef je als argument mee. Deze moet in de Form directory staan!

De methode createView geeft een object die gebruikt kan worden in een twig view. De twig functies form\_start, form\_widget en form\_end maken hier gebruik van.

Hieronder de template admin/new\_afdeling.html.twig

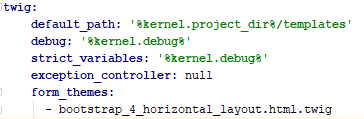


Voer de actie in de browser uit

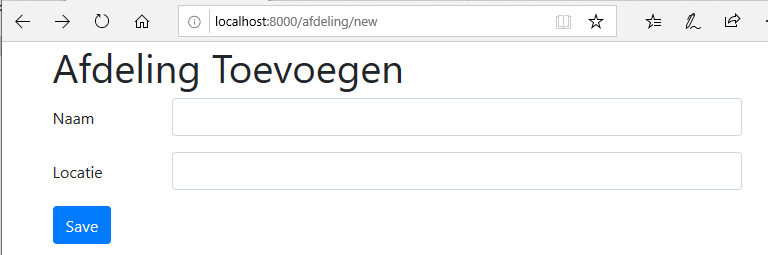


Dit formulier gebruikt de default stylesheets van Symfony om het formulier op te maken.

Voeg het volgende aan config/packages/twig.yml toe:



Nu gebruikt Symfony de bootstrap stylesheet. Zoveel mooier. Let op, wel zelf de css en js map vullen met bootstrap!



De gegenereerde HTML

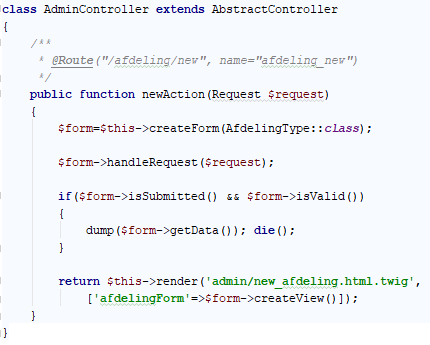


Wat opvalt is dat er geen action attribuut aan het formulier is toegevoegd. Na het posten wordt dezelfde action /afdeling/new opnieuw uitgevoerd. Echter nu wel met een aantal POST variabelen:

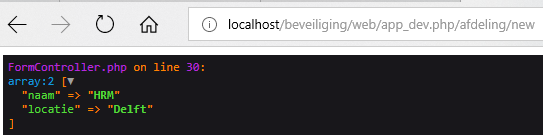
* afdeling[naam]
* afdeling[locatie]
* afdeling[\_token]

De laatste variabele wordt ‘hidden’(niet in je formulier zichtbaar) verstuurd en zorgt ervoor dat Cross-Site Request Forgery (CSRF) niet mogelijk zijn. Kortweg betekent dit dat alleen formulieren verstuurd door de server eenmalig teruggestuurd kunnen worden.

De action is verantwoordelijk voor het renderen van het formulier **en** het verwerken van het formulier.



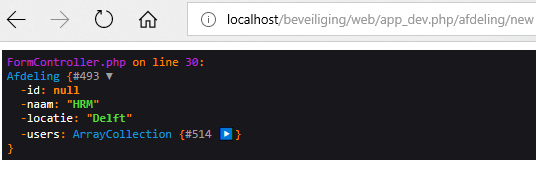
Voer de action uit, vul het formulier in en submit deze.



De ontvangen data is een associatieve array met twee naam-waarde paren. Nu kunnen we een afdelings-object maken en deze vullen met de ontvangen data en daarna opslaan in de database.

Maar waarom laten we Symfony niet deze stap uitvoeren…



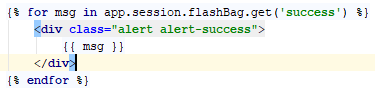
Vul het formulier in en submit deze. Nu krijgen we een Afdeling object terug.

Nu het object opslaan en klaar.



Je ziet dat deze action twee afwerkingen kent

1. Het formulier is niet gepost en wordt alleen gerenderd.
2. Het formulier wordt gepost
3. Controleer of de geposte data voldoet aan de validatie regels in de entiteit (validation)
4. Vraag het ingevulde afdelings object op en bewaar deze in de database.
5. Maak een Flash, bewaar een mededeling in de session. De volgende pagina kan deze mededeling weer opvragen en tonen. Zo ziet de gebruiker dat er een afdeling succesvol is aangemaakt.
6. Altijd redirecten naar een nieuwe pagina (action), toon de flash in de twig file met



Standaard maakt Symfony van elk invoer veld een type=”text” veld. Is het formulier gekoppeld aan een entiteit dan wordt ‘Form Type Guessing’ toegepast

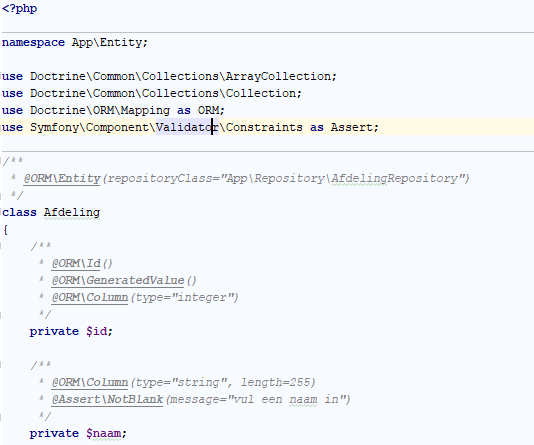
Wil je dit anders dan moet je dat aangeven in je formulier.





De locatie is een select geworden.

Willen we nu dat de naam van de afdeling altijd ingevuld wordt dan moeten we dat aangeven in de Afdeling entity



Voeg je geen naam toe dan volgt er een foutmelding $form->isValid() resulteert in false.

De foutmelding wordt in je formulier getoond. Symfony zorgt dat de oude formulier velden hun geposte data weer krijgen. Dus niet het hele formulier opnieuw invullen! (ik heb Naaldwijk geselecteerd)

Middels validation regels kan je het gehele formulier beschermen. Nice.

