Vervolg tutorial Slim framework

# Introductie

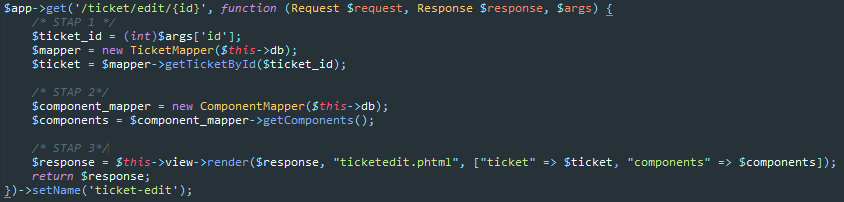
Deze tutorial is een vervolg op de officiële tutorial van het Slim framework genaamd “First Application”. Ik vond dat er nog enkele belangrijke dingen vergeten waren dus maak ik een klein vervolg op de tutorial.

# Tutorial

Nadat we ervoor gezorgd hebben dat we tickets kunnen weergeven, aanmaken en de details ervan kunnen zien, gaan we nu nog een edit pagina en een delete functie aanmaken.

## Edit

We beginnen met de edit functie van een ticket te maken. Het eerste dat we daar voor moeten doen is een route aanmaken zodat als je naar /ticket/edit/{id} surft het programma weet welke template hij moet weergeven en welke functies het moet uitvoeren. We geven met een GET variabele het ID van het ticket dat we moeten bijwerken mee zodat we in de database kunnen zoeken op het ID en het juiste ticket er uit kunnen halen.



De eerste stap dat je moet doen is ervoor zorgen dat je het juiste ticket uit de database kan halen omdat je de data van het ticket moet kunnen weergeven op de edit pagina. Je haalt eerst het ID van je ticket uit de URL zodat je daarna met de methode getTicketById() van de TicketMapper (klasse waar alle methodes in staan voor bewerkingen uit te voeren van een ticket) het juiste ticket ophaalt en in de variabele $ticket steekt.

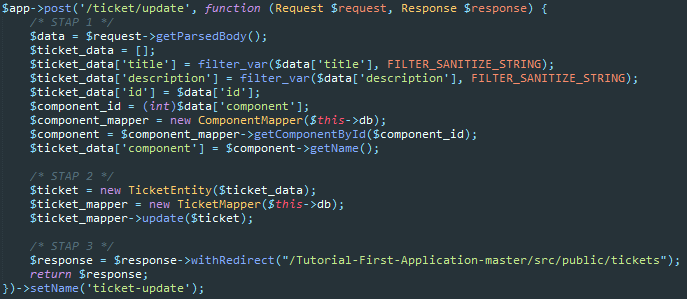
Bij de tweede stap moet je ervoor zorgen dat je alle mogelijke componenten kunt weergeven. Dat doe je door gewoonweg de componenten op te vragen door de methode getComponents() uit te voeren van de ComponentMapper. De uitkomst van die methode sla je dan op in de variable $components.

Als laatste stap geef je mee welke template de applicatie moet laten zien, in dit geval is dat ‘ticketedit.phtml’. Je moet dan ook nog de variabelen ticket en components meegeven, want als je dat niet doet kan je die gegevens niet gebruiken in de template. We zetten de naam van de route nog op ticket-edit, op die manier kunnen we in de templates gemakkelijk verwijzen welke route de applicatie moet gebruiken.

Nadat je de route hebt opgesteld moet je de template ticketedit.phtml natuurlijk nog aanmaken. Een voorbeeld van deze template staat hieronder.



Eens we de waarden die we willen veranderen hebben aangepast kan je klikken op de edit button, de form zal je naar ticket/update sturen en daar zal de data worden aangepast. We hebben dus een nieuwe route nodig.

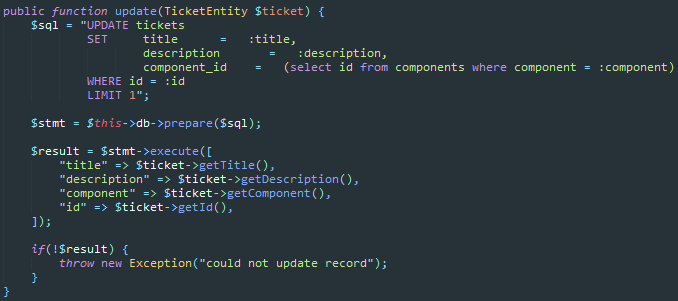


Bij de eerste stap halen we de data die we ingegeven hebben op. We halen titel, omschrijving, component naam en id van het ticket op.

Bij stap twee maken we van die data een ticket entiteit en steken die in een variabele $ticket. Die variabele gebruiken we dan in de methode update() die we zo dadelijk nog gaan maken.

Als het ticket is geüpdatet zorgt de laatste stap ervoor dat we worden terug gebracht naar het overzicht van alle tickets. Ook deze route geven we nog een naam mee.

De nieuwe update methode maken we waar alle andere ticket methodes staan en dat is in de TicketMapper klasse.

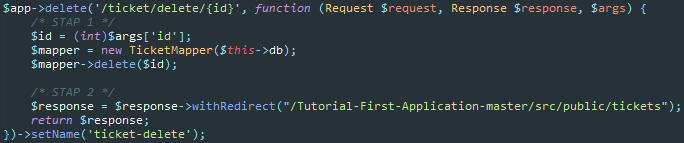


Het eerste dat we moeten doen is de mysql query typen die we gaan uitvoeren om de data aan te passen. Eens dat gebeurd is moeten we nog zeggen welke data bij welke variabele hoort en het spreekt voor zich dat $ticket->getTitle() bij title hoort, $ticket->getDescription() bij description hoort, enzovoort. We geven de methode ook nog 1 argument, dat is de data van het ticket dat je wilt aanpassen.

Als alles goed is verlopen kan je nu tickets bijwerken.

## Delete

Net zoals bij de edit functie moeten we bij de delete functie een route aanmaken waar we het id van het ticket meegeven dat wel willen verwijderen.



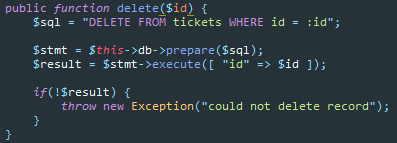
Bij de eerste stap vangen we het id op van het ticket dat wel willen verwijderen en steken we die in een variabele $id. Daarachter gebruiken we de delete methode, die we direct gaan aanmaken, van TicketMapper om het ticket te verwijderen.

Eens dat allemaal in orde is gaan we bij de laatste stap de gebruiker terug sturen naar het overzicht van allen tickets. Ook hier gaan we de route een naam geven.

Zoals je ziet gebruiken we hier de delete route, daarom moeten we zien dat we in de template een form maken dat daarop is afgesteld. Een voorbeeld van zo een form staat hieronder.



Als dat allemaal in orde is kunnen we beginnen met de delete methode. Deze is iets korter als de update methode.



Net zoals bij de update methode moete we eerste de query die we willen uitvoeren maken. Als dat gebeurd is moeten we zeggen dat het id van een ticket moet overeenkomen met het id dat we hebben meegegeven.

En dat is het, als ook hier alles goed is verlopen kan je nu een ticket verwijderen.