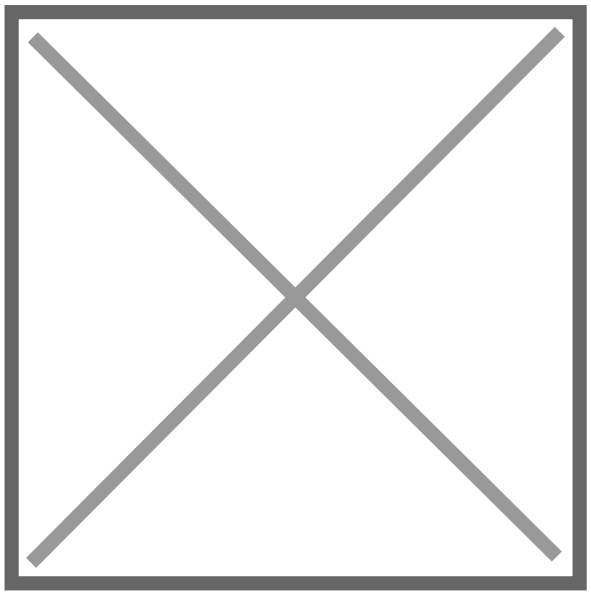
5.2 CRUD - Create

We hebben de R in de vorige les gemaakt. We gaan nu de C van create maken. Daarmee kunnen we een Airline toevoegen.

We maken hiervoor een file insert.html en in deze file maken we een form waarin de gebruiker wordt gevraagd om de gegevens van de nieuwe ailine in te typen. Een nieuwe airline bestaan slechts uit twee velden: de airline code en de airline name. Het form ziet er dus ongeveer als volgt uit:



Een eenvoudig form met twee velden en een button. De button post de velden naar de file insert.php. Deze insert.php voert dan de juiste query uit gebrukt daarvoor het object DB.

We hebben dus een rijtje van drie files die elkaar aanroepen: het form vraagt de waarde aan de gebruiker, de php file ontvangt de waarden uit het form en stuurt deze in de juiste vorm naar insert.php. Insert.php roeptde crud.php aan waar de juiste query wordt uitgevoerd.

File	Functie	MVC
insert.html	Formulier, GUI naar gebruiker	View (=html, GUI)
insert.php	De code waar wordt bepaald wat er wordt gedaan. Voor een insert is dat echter vrij eenvoudig.	Control (=code)
crud.php	De vertaling naar een query naar de database. CRUD maakt zelf weer gebruik van DB om een verbinidng naar de DB te maken. Dit is allemaal nodig om met de Database te communiceren.	Model (=Database)

Waarom deze 'moeilijke' scheiding in allemaal verschillende file, zul je misschien afvragen? Dit zullen we in de les verder uitleggen, maar het komt er op neer dat je alle functies zoveel mogelijk wilt scheiden. Dit komt allemaal ten goede aan code die goed onderhoudbaar is. Stel je maar eens voor dat je over en nergens een SQL query kun tegenkomen en de datanase tabelnamen veranderen, dan weet je niet waar je dit allemaal moet aanpassen.

OK we gaan de code aanpassen, we beginnen met crud.php.

```
<?php
...
    public function insertNewAirline($code, $name) {
        return $this->executeSQL("insert into airlines (code, airline) values ('$code',
'$name')");
    }
...
```

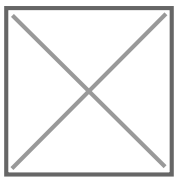
We voegen een nieuwe method to die een rij in de database toevoegd. De rows in de tabel in dit voorbeeld heten code en airline en de tabel heet airlines. Dat kan in jou database anders heten. De method verwacht twee parameters, $code en $name. Maar een file insert.php en zorg deze method wordt aangeroepen en dan dat de geposte variabelen uit het form als parameters worden meegegeven.

Let op dat als je een method aanroept dat je dan wel het jusite object eerst moet instantiëren. Kijk daarvoor nog eens terug hoe je dat hebt gedaan in view.php

Als de insert is gelukt vanuit insert.php dan ga je vanuit insert.php terug naar view.hp. Je kunt daarvoor de PHP functie header("") gebruiken. Deze stuurt een header naar je browser en daarin kun je dan aangeven welke pagina moet worden geladen. Zoek zelf op hoe dit werkt.

Als het goed werkt dan zul je zien (via de view.php) dat je een regel heb toegevoegd.

Je kunt nu ergens in je view.php een <a href....> plaatsen zodat je vanuit deze pagina naar de insert kunt springen.



Wat gebeuert er nu als je een Airline code voor een tweede keer probeert in te voeren?

Je krijgt een foutmelding, waar komt die vandaan?

Opdracht/Huiswerk

1. Zorg ervoor dat de input wordt gecontroleerd. Een airlinie code moet altijd uit twee hoofdletter bestaan en de Airline naam mag niet leeg zijn.
2. Pas je code nu zodanig aan dat je een nette foutmelding krijgt als je een Airline code voor een tweede keer probeert toe te voegen.
   Je zult hiervoor de control (code) moeten aanpassen. Welke file is dat? Bedenk eerst hoe je dit wilt gaan aanpakken en overleg met de docent.

Stuur de file insert.php op via teams als controle voor het uitvoeren van het huiswerk. Dit wordt afgetekent.

--

Revision #6
Created 2019-09-23 12:55:13 UTC by Admin
Updated 2020-10-05 11:17:56 UTC by Max