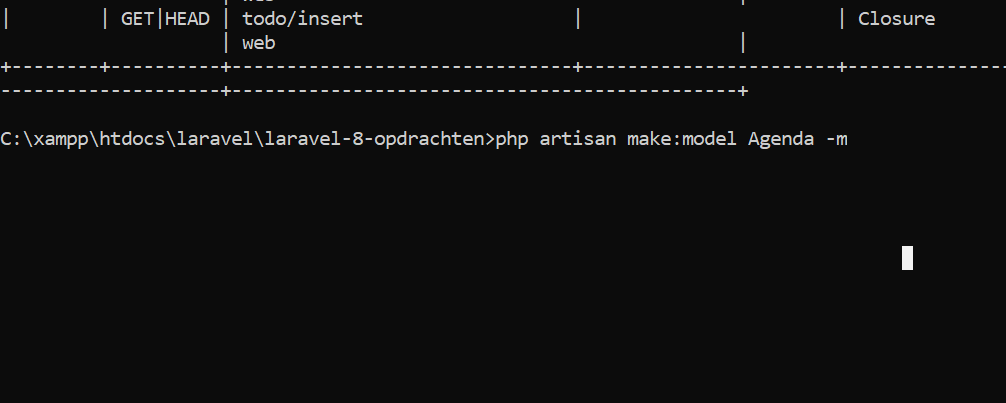
Crud Acties uitwerken: Agenda

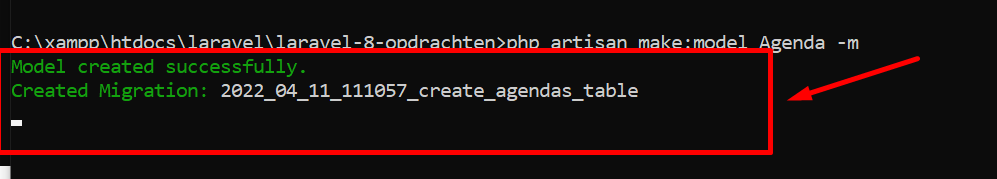
Doelen na de les: uitwerken van de Create, Read, Update, Delete op de tabel Agenda.

Opdracht: na de les complete crud acties af hebben vandaag.

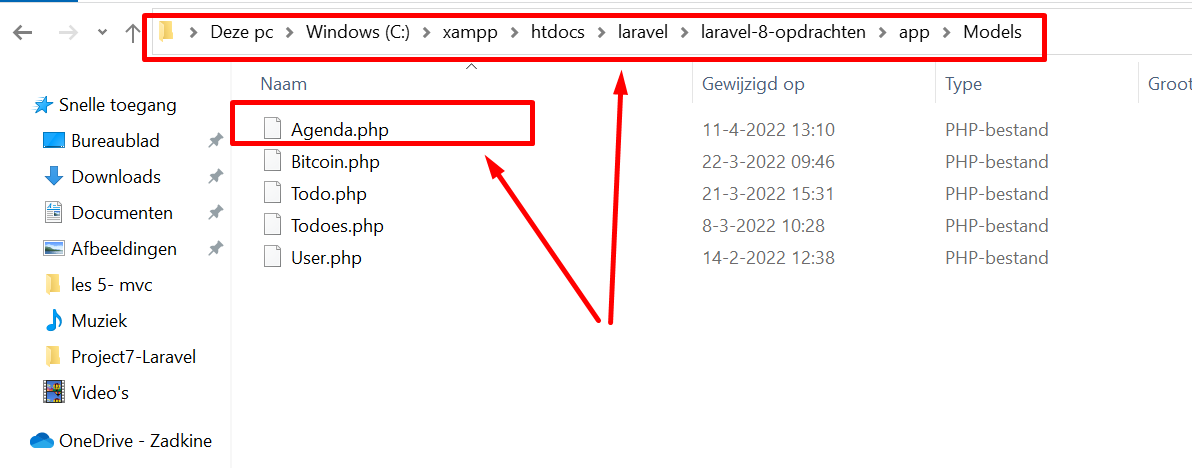
Model Agenda Maken



Als we dat uitgevoerd hebben:



Hoe gaan we dat verwerken in Laravel en waar zit het Model Agenda ook alweer?



Hier doen we nog even niets mee.

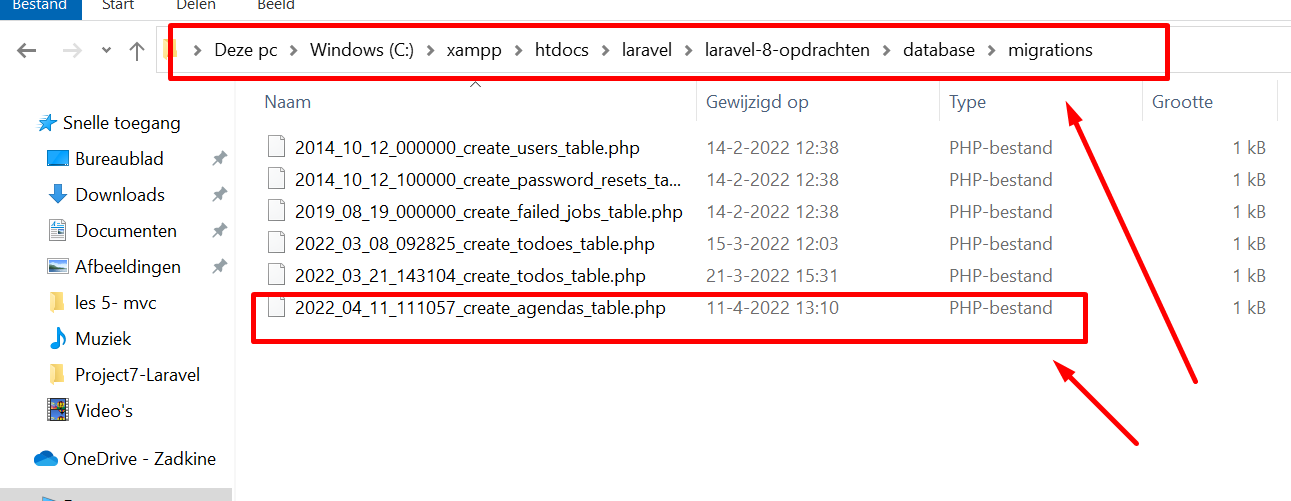
We gaan aangeven aan Laravel hoe wij de tabellen voor Agenda willen hebben:

Dit is een modelschets voor hoe we de tabellen willen hebben.

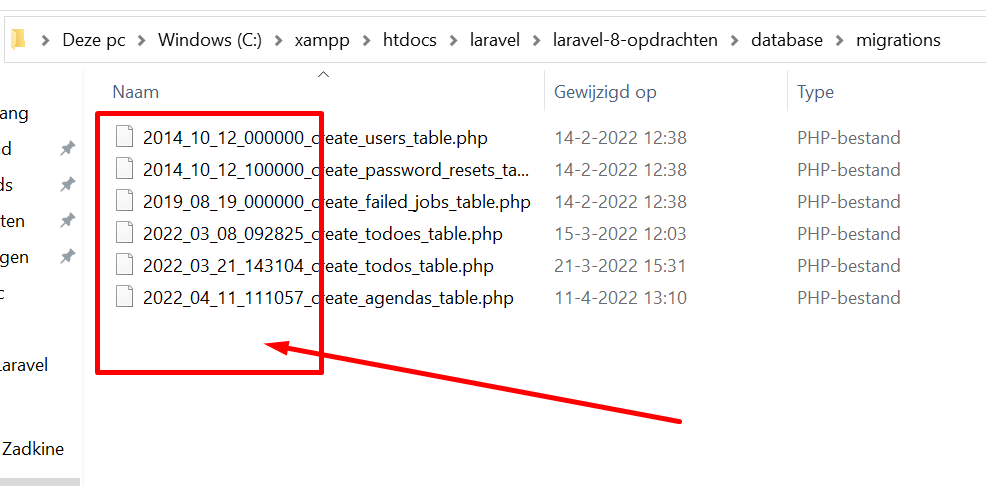
Tabelnaam: agenda

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **kolomnaam** | **type** | **Max** | **Null** |  | Opmerkingen |
| id | integer | 11 | Nee |  | Standaard door Laravel gedaan voor je, hoef je niet perse op te geven. |
| naam | string | 250 | nee |  |  |
| beschrijving | Text |  | nee |  |  |
| actief | tinyint | 1 | ja | Default op 1 |  |

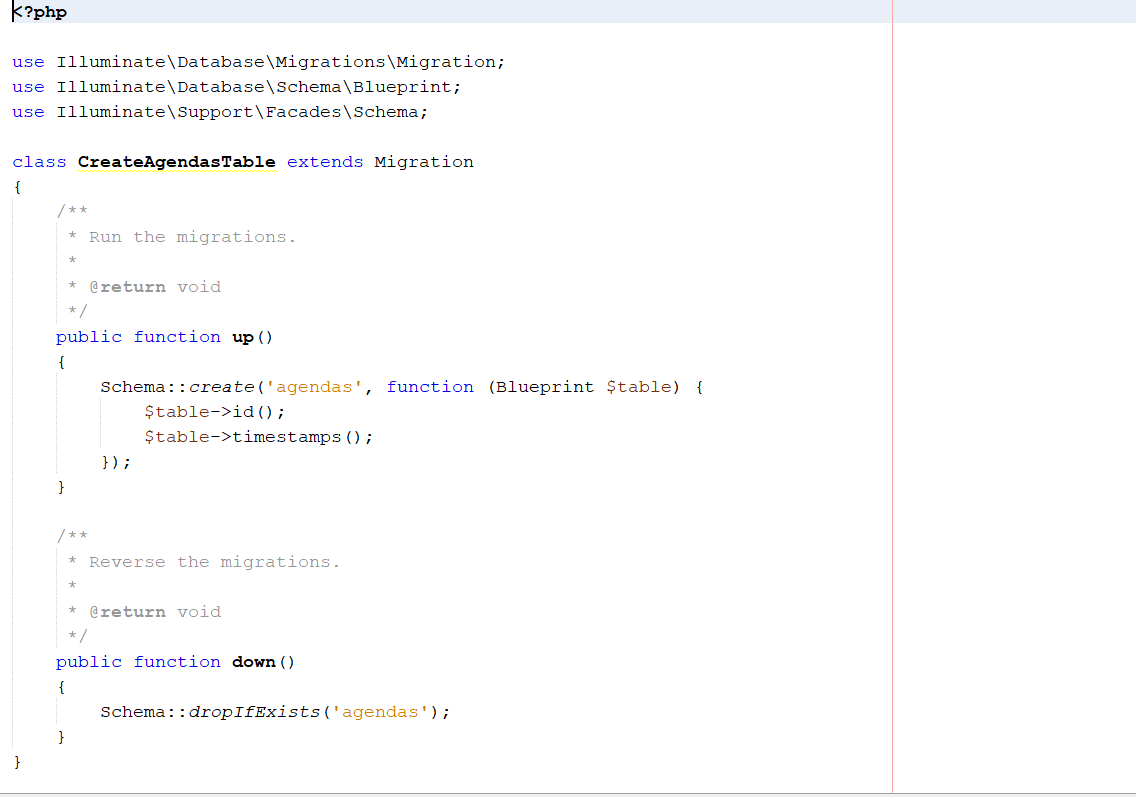
Hiervoor moeten we in Laravel een nieuwe bestand zoeken in folder zie afbeelding:



LET OP, zie afbeelding hieronder, rood gemarkeerd is een aanmaakdatum, dus deze verschilt bij jullie:



Open het bestand: <datum> \_create\_agendas\_table .php



Pas het aan naar:



Tekst:

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('agendas', function (Blueprint $table) {

$table->id();

$table->string("naam", '250');

$table->text("beschrijving");

$table->tinyInteger('actief')->default(1)->nullable(true);

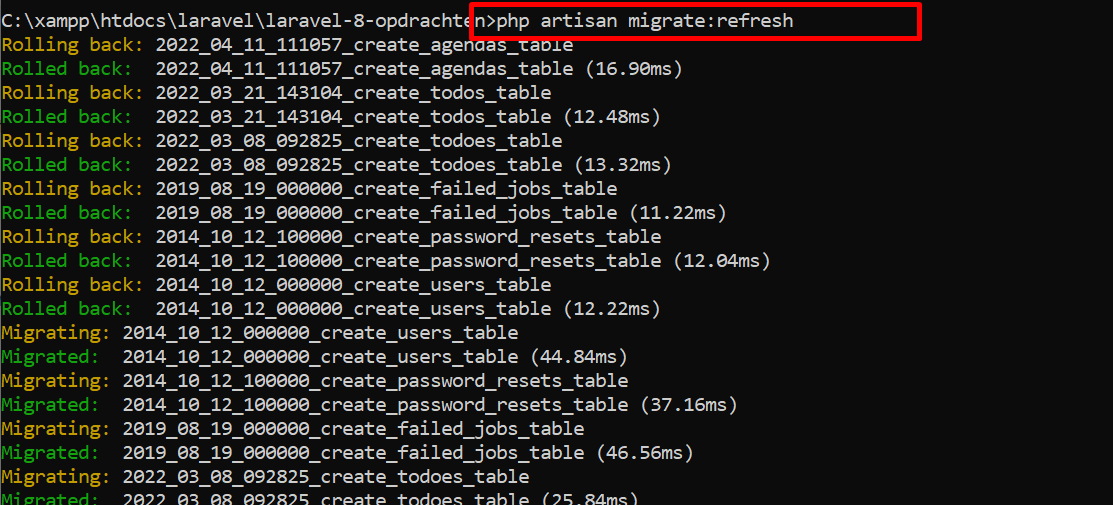
$table->timestamps();

});

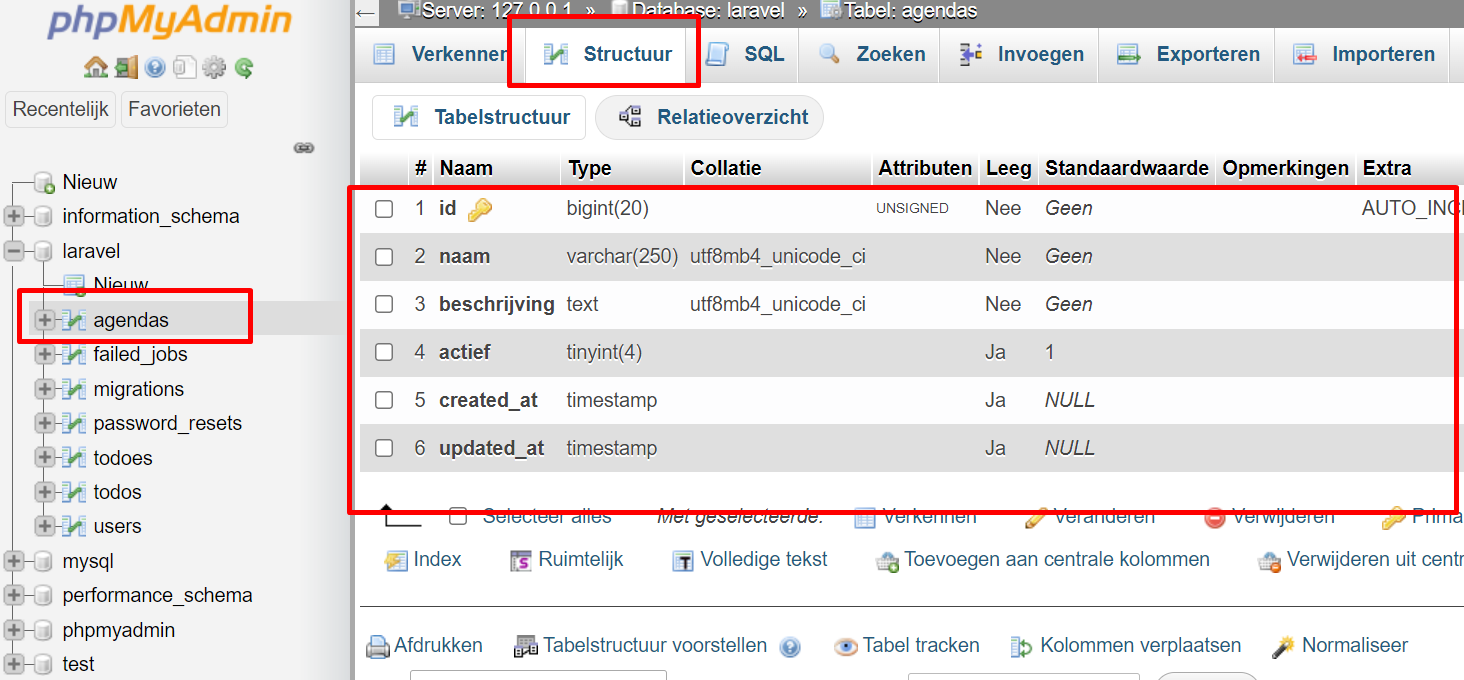
}

// Sla het bestand op

Voer het volgende commando in: php artisan migrate:refresh



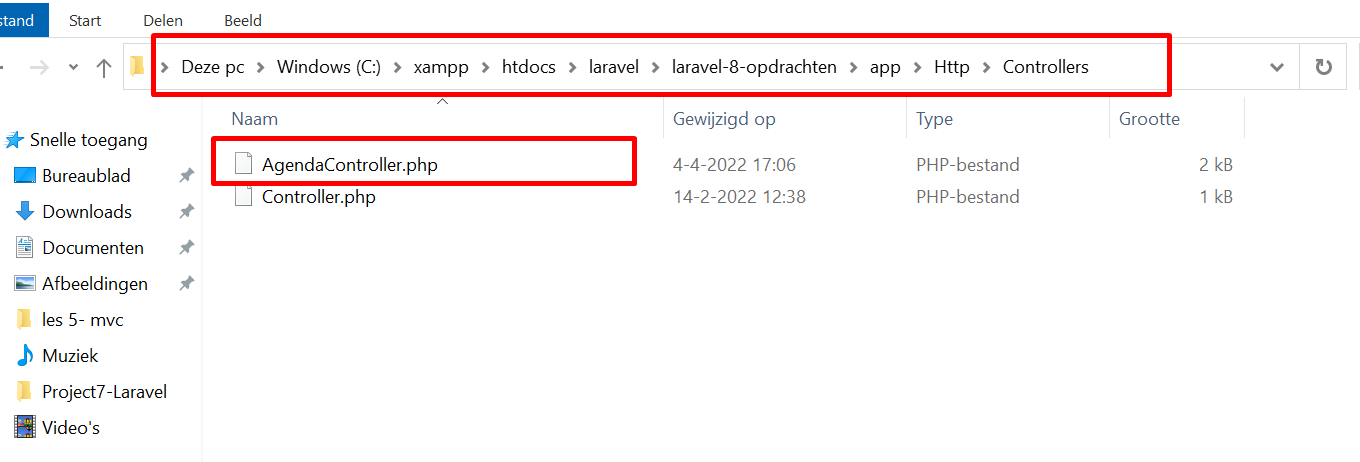
Kijk nu in je PhpMyAdmin:



DEEL 2 – CRUD acties verwerken

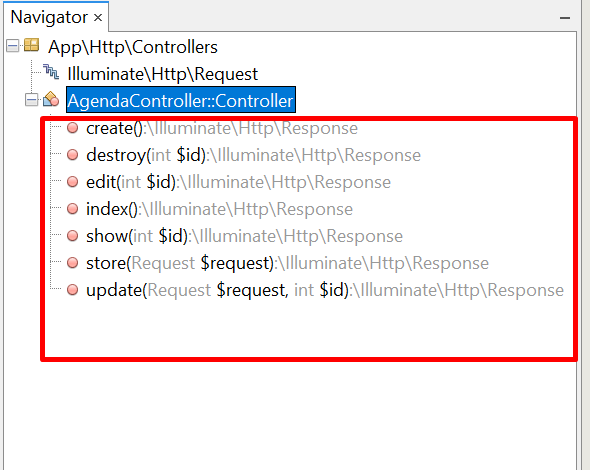
Create aanmaken en verder koppelen in je controller:

Open je controller bestand voor Agenda:



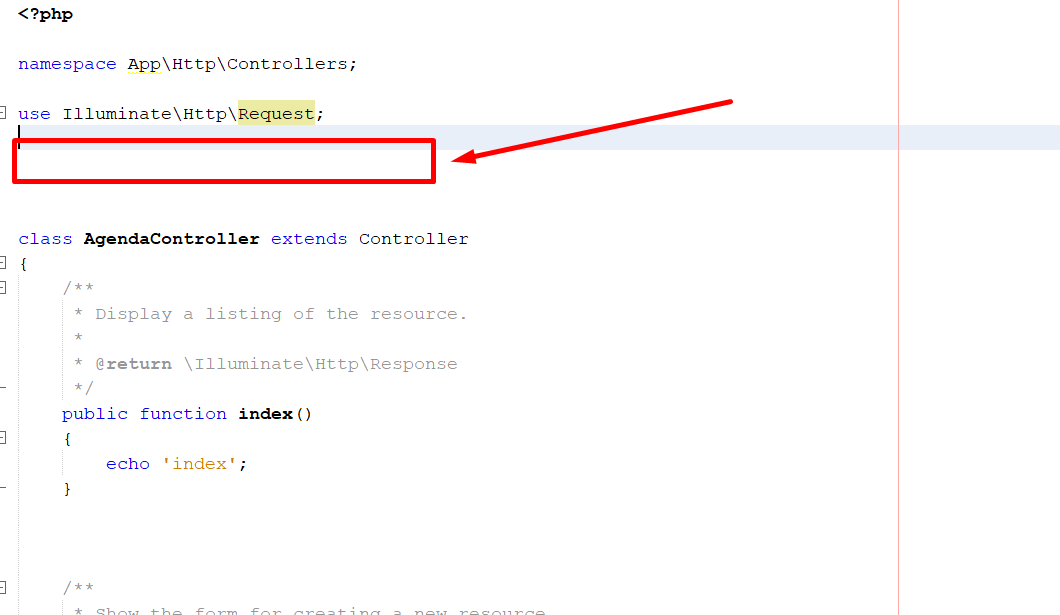
Open het bestand in je editor naar keuze:

Hier zitten de volgende functies al in, dit is al door Laravel voor je klaargemaakt.

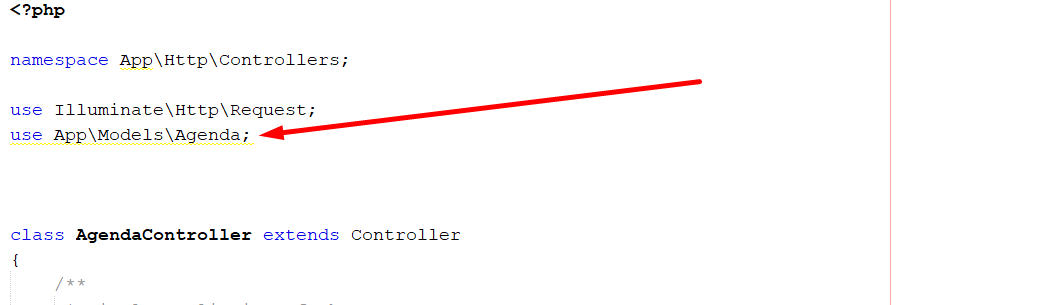


We gaan hier het model ‘Agenda’ in opnemen als een ‘namespace’.

Zet volgende regel erbij: use App\Models\Agenda;

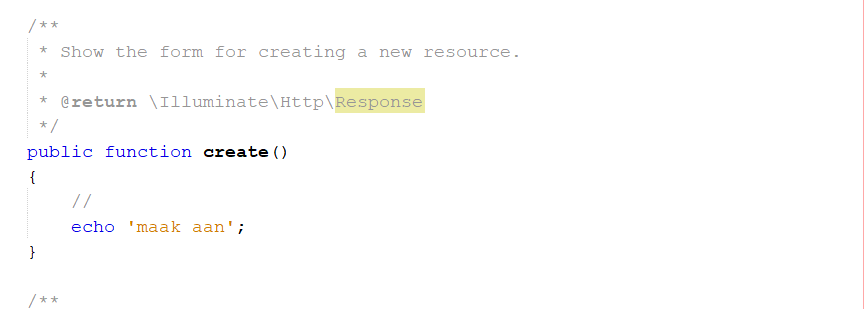


Zie hier een voorbeeld met de namespace er in:



We gaan een create maken.

In de vorige opdrachten hebben we hier alleen een echo ingezet:



Hoe werkt een create? We zetten in ons voorbeeld zelf eerst een data array op. Die gaan we wegschrijven naar de tabel. Klinkt spannend maar super eenvoudig met Laravel.

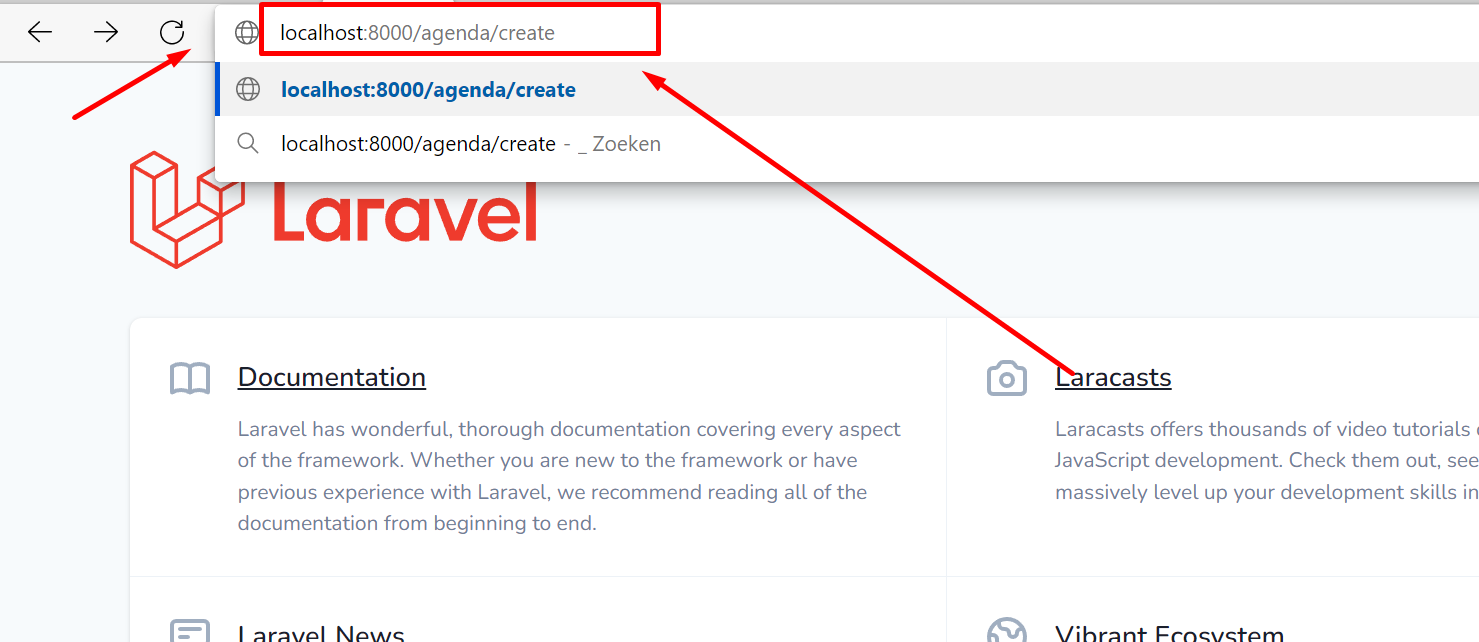
Normaal heb je altijd een formulier op een webpagina, waar je data invult, en die valideert en vervolgens wegschrijft (asl het goed is).

In ons voorbeeld schrijven we direct naar een tabel, we bouwen dit steeds uit.

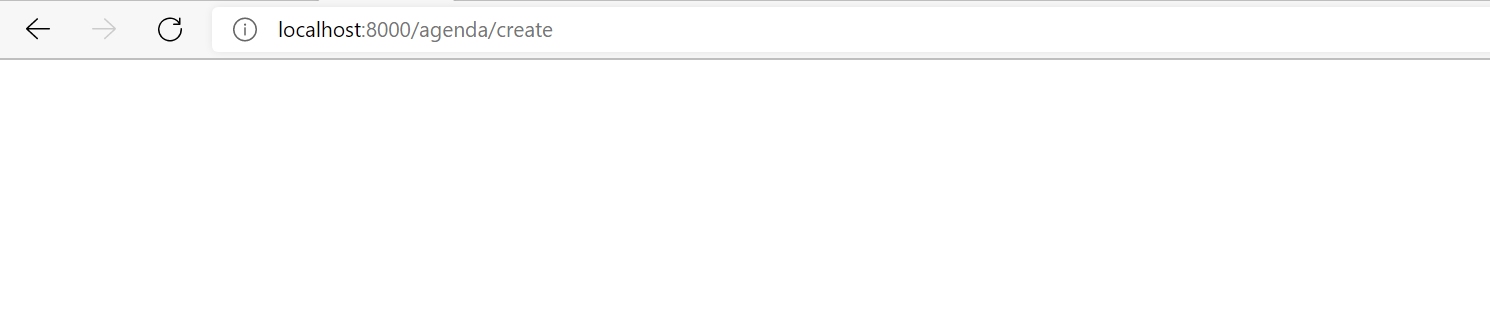
Create functie aanpassen naar:



* Open een webbrowser
* Zorg dat laravel opgestart is: php artisan serve
* Ga naar de create route: zie hieronder



Er is verder geen code opgegeven dus de pagina is leeg verder



* Open je phpmyadmin



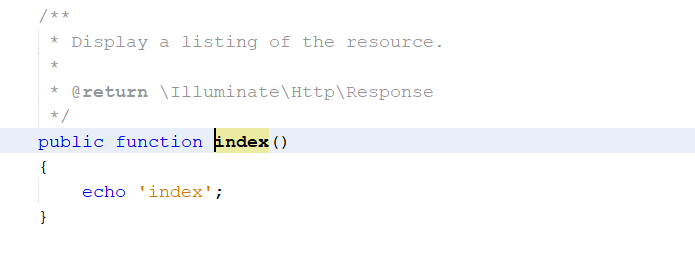
Zie hier een insert.

Create is hiermee nu afgerond, later gaan we dit mooier maken.

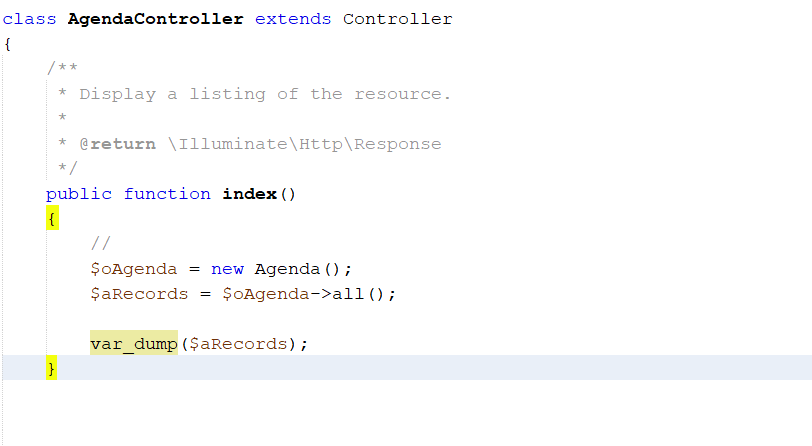
Read route:

We gaan de tabel in zijn geheel uitlezen:

Verwerk dit in de ‘index’ functie”



Pas dit aan naar:

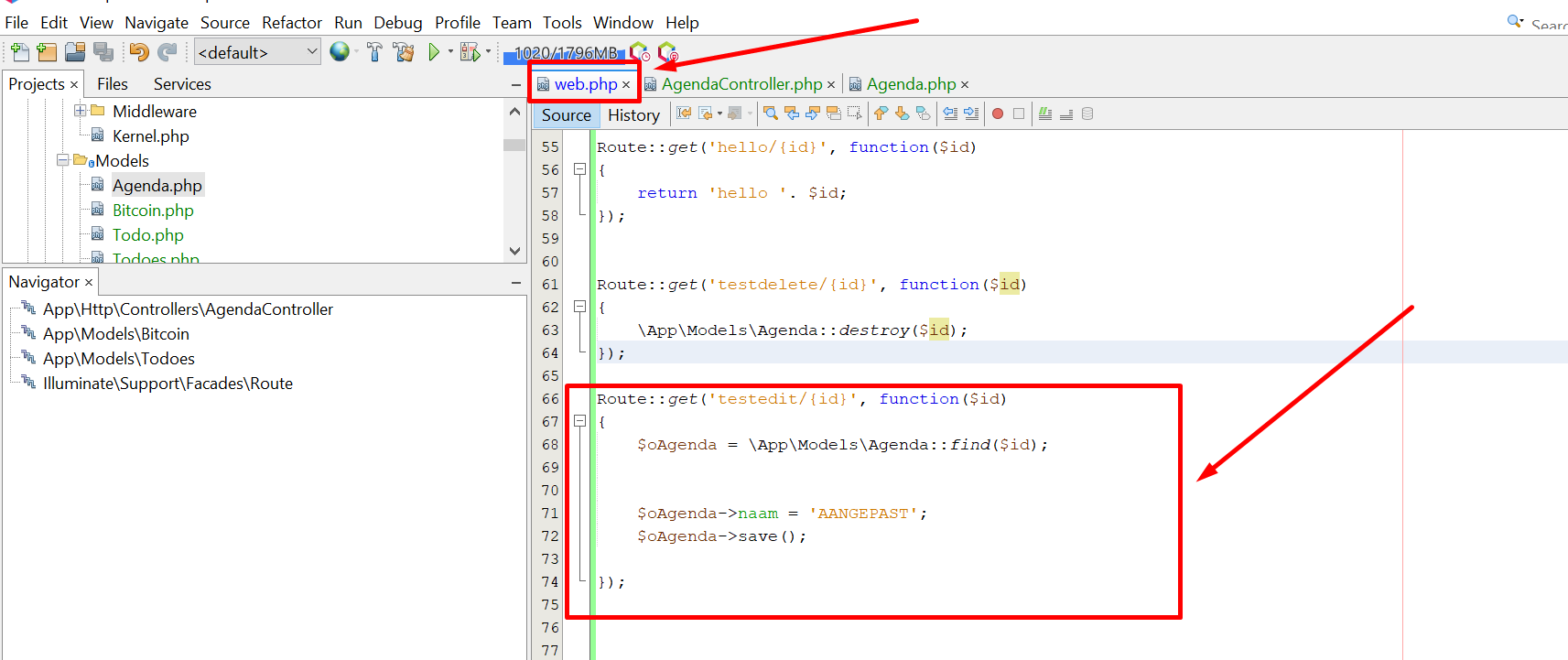


* Open een webbrowser
* Ga naar de route: localhost:8000/agenda
* Omdat het een index functie is, hoef je dit niet perse op te geven in de route (localhost:8000/agenda/index == gelijk aan localhost:8000/agenda)

UPDATE ROUTE

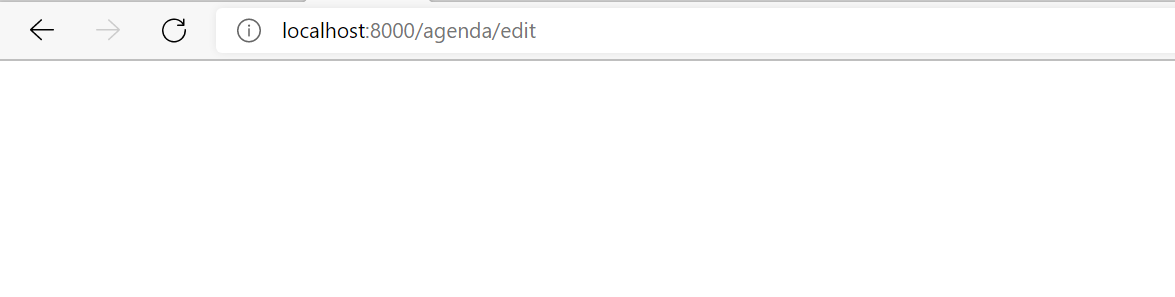
Maak je ‘edit’ functie als de afbeelding hieronder:

LET OP: we maken deze route tijdelijk in ‘web.php’

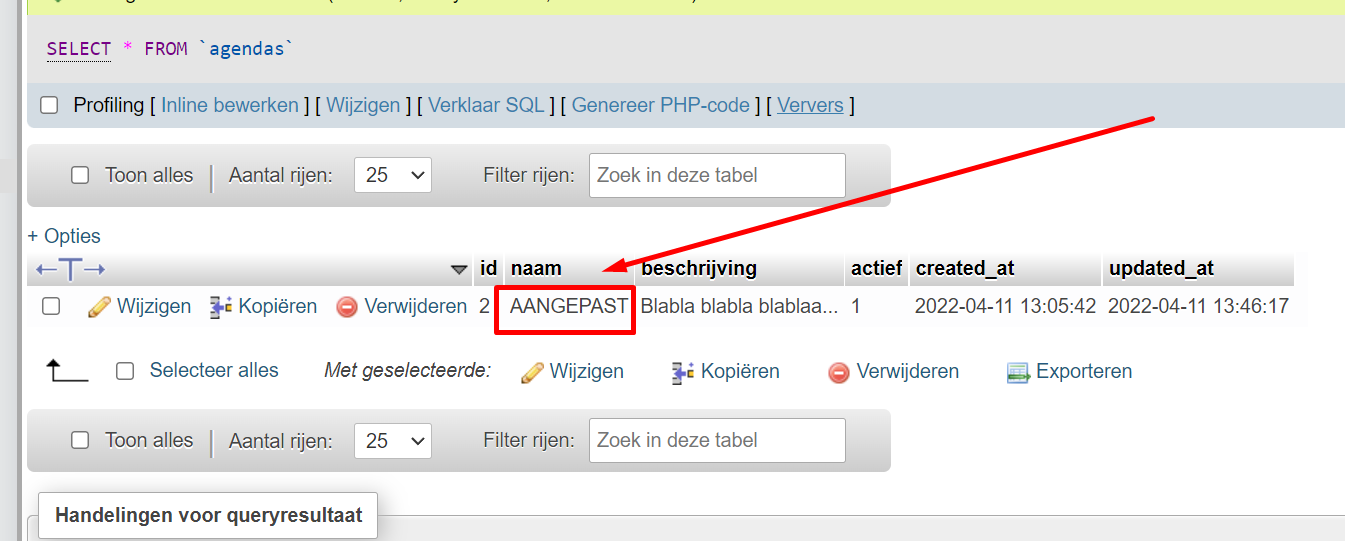


* Open een webbrowser
* Ga naar de route: localhost:8000/agenda/edit

Leeg is goed; je hebt in ieder geval geen fouten dan 😊

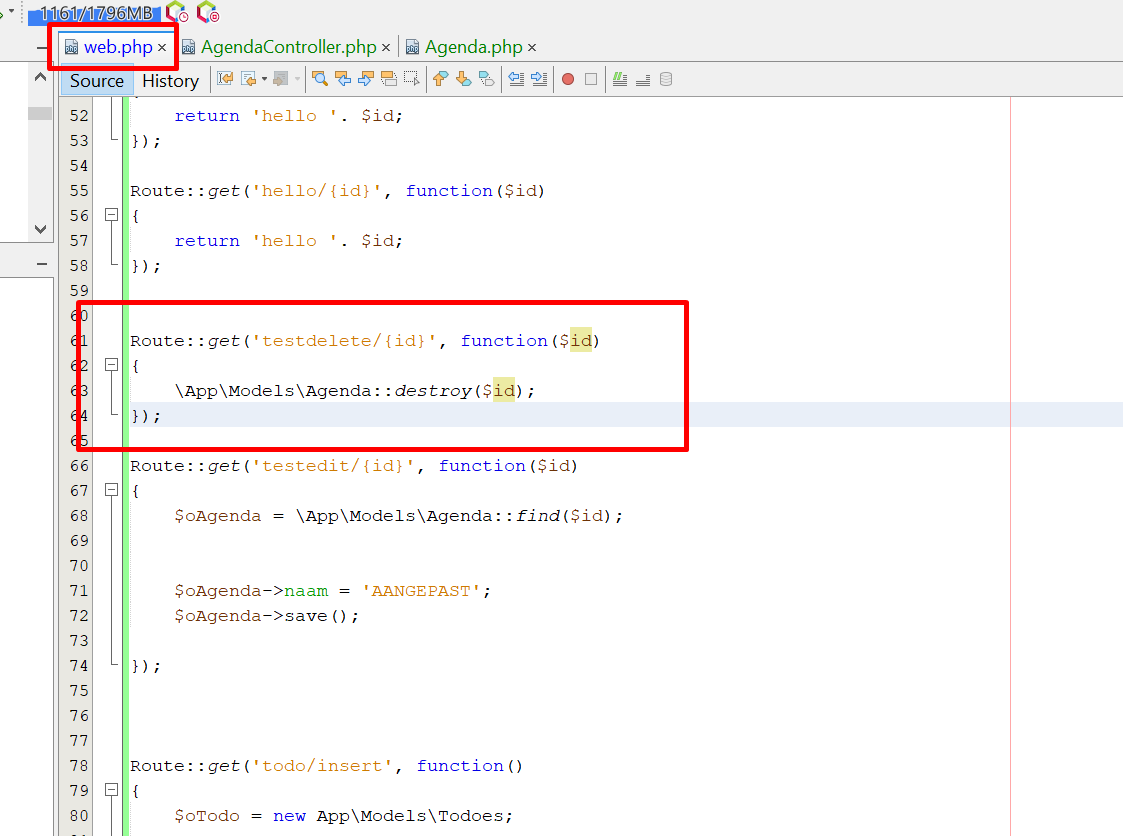


* Open je phpMyAdmin



Delete route maken

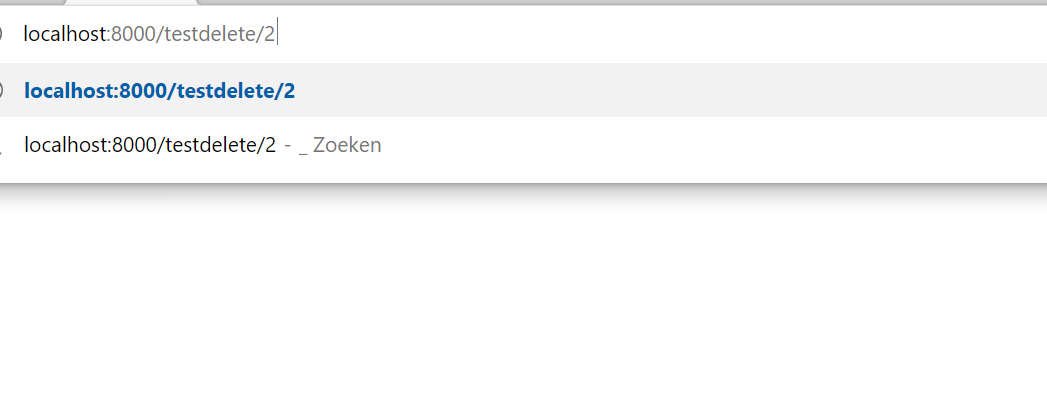
We maken deze route (tijdelijk) in ‘web.php’.



Voor het verwijderen hebben we altijd een <ID> nodig van het te verwijderen record.

We geven hier nu zelf ‘1’ aan als record met id = 1.

Wanneer je deze verwijdert, dan bestaat deze niet meer! Kijk goed in je phpmyadmin welke records er zijn met welke id’s . Dit moet je voor nu even in de code dan zelf wijzigen.



Hiermee hebben we nu 4 routes gemaakt die alle 4 moeten werken.

Create, Delete, Update en Read routes.

*(\*op de update en delete zijn de routes voor nu even iets anders uitgevoerd.*)