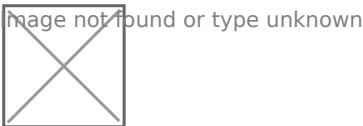


# Overzicht van URL naar view met 1:M relatie

*Hieronder een schematisch overzicht van de belangrijkste onderdelen hoe je van een URL via de controller en model de juiste informatie in de view plaatst.*



1. Dit is de URL die de gebruiker invoert. Via de [routing](#) worden twee items uit de URL gehaald: (a) *country* en (b) *overzicht*. Item (a) is de controller-naam en item (b) is functie-naam (methodenaam). De controller zelf is te vinden in de file `controllers/Country.php` deze file-naam komt ook overeen met de item (a).
2. In de methode `actionOverzicht`, waarvan de naam dus uit item (b) van de URL komt, wordt een `$country` object gemaakt. Dit object is verbonden met de methode `Country` en deze is weer beschreven in de `models/Country.php` en wijst vanuit daar naar de juiste database tabel.
3. Vanuit de controller wordt er verwezen naar het model. In het model is beschreven waar de data werkelijk kan worden gevonden. In de methode `tableNaam()` wordt beschreven in welke tabel de informatie staat. In de methodes `getCapital` en `getCountrylanguages` wordt beschreven hoe een andere tabel is gekoppeld aan deze country tabel.
4. In de view kunnen we het object `$country` dat is gemaakt in de controller gebruiken om gegevens uit de tabel af te drukken. Via de methodes `getCapital` en `getCountrylanguages` kunnen gegevens uit de andere tabellen die zijn gekoppeld aan `$country` ook worden afgedrukt. Afhankelijk van de soort relatie 1:1 of 1:N krijg je een waarde of een array met waardes terug.
5. De view maakt (rendert) de hele pagina en stuurt deze in zijn geheel naar de browser.

*Opmerking: Een methode (NL) of method (EN) is een functie in een class. Zodra je leest `public function Abc() {` dan is Abs eigenlijk een methode.*

# Toelichting Stap 3a

```
public function getRelatie() {  
    return $this->hasMany(Relatie::className(), ['key.relatie' => 'key.thismodel']);  
}
```

---

Revision #9

Created 12 July 2020 11:55:56 by Max

Updated 7 December 2020 21:36:31 by Max