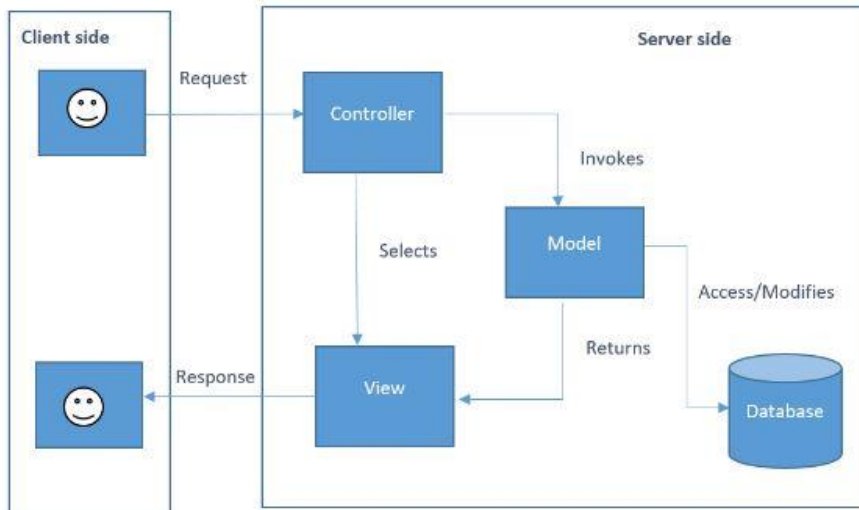


# ASP.NET Core MVC Web app

## 1 Inleiding

### 1.1 Overzicht MVC Architectuur



**MVC** - is een architecturaal pattern

Structuur bestaat uit 3 componenten (models / controllers / views)

3 componenten die los aan elkaar hangen (loose coupling) (voordelig voor maintenance/testing)

Dit is het **SoC principle (Separation of Concerns)**

Elke component heeft zijn eigen verantwoordelijkheid:

**Model** : data (lezen/schrijven/wijzigen) bv via een ORM (Object Relational Mapper) zoals EF Core

**View** : verzorgt de UI (representatie van gegevens) die hij meekrijgt van Controller

(naar User HTTP response request) - web pagina in browser

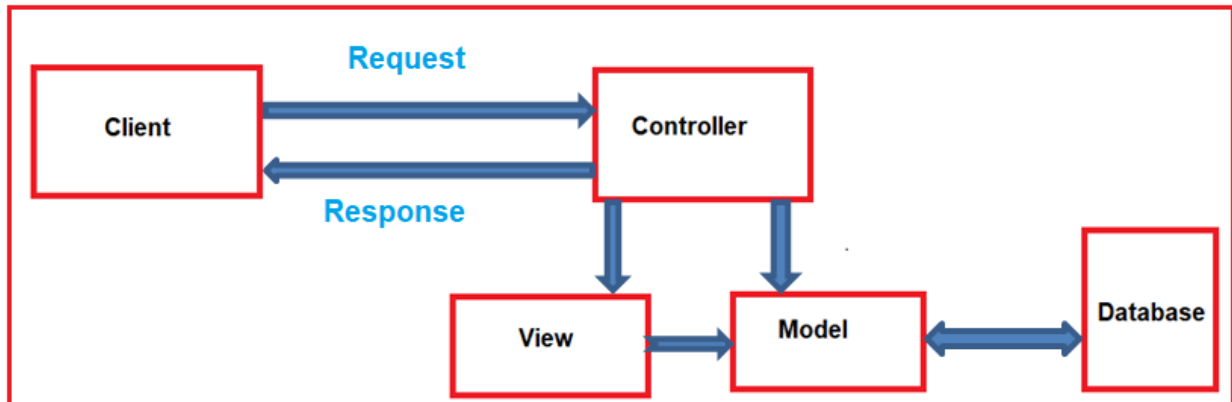
**Controller**: Requests afhandelen (HttpGet/HttpPost)

via (action) methodes

daarvoor gaat hij Model vragen om data en ev.

Zelf data aan model leveren om naar de db te schrijven

Geeft gegevens door aan View die moeten worden getoond aan eindgebruiker



## 2 Referenties

<https://asp.mvc-tutorial.com/introduction/what-is-mvc/>

<https://dotnettutorials.net/lesson/introduction-asp-net-core-mvc/>

## 2.1 Routing in mvc

Bv <https://www.website.com/Customers/Edit/101>

Standaard gedrag: Zal een methode Edit(...) aanroepen in de CustomersController class en een parameter (meestal een id van type int ) met waarde 101 meegeven aan deze methode Edit

### Routing systeem:

Request wordt doorgestuurd naar de CustomersController

En Methode Edit in CustomersController wordt dan uitgevoerd

Indien de gevraagde request url niet wordt herkend, wordt een Foutcode 404-Page not found teruggegeven als http response naar de browser van de eindgebruiker

