Applicatiebeheer

MaRoWo Case 2

***Gemaakt door:***

*Max Mommersteeg*

*Rob Gerritsen*

*Wouter Fennis*

***Datum:*** *01-12-2016*

Inhoud

[Ontwerp 3](#_Toc468367433)

[Case2.MaRoWo.GarageAdministratie.FrontEnd (GA.FE) 3](#_Toc468367434)

[Onderdelen 3](#_Toc468367435)

[Case2.MaRoWo.OnderhoudBeheer.Service (OB.Service) 4](#_Toc468367436)

[Onderdelen 4](#_Toc468367437)

[Inkomende commands 5](#_Toc468367438)

[Inkomende events 5](#_Toc468367439)

[Uitgaande events 5](#_Toc468367440)

[Case2.MaRoWo.RDW.IntegrationService (RDW.IS) 6](#_Toc468367441)

[Architectuur 6](#_Toc468367442)

[Inkomende commands 6](#_Toc468367443)

[Uitgaande requests 6](#_Toc468367444)

[Uitgaande events 6](#_Toc468367445)

[Development 7](#_Toc468367446)

[Testhandleiding 8](#_Toc468367447)

[Troubleshooting 9](#_Toc468367448)

# Ontwerp

## Case2.MaRoWo.GarageAdministratie.FrontEnd (GA.FE)

Een ASP.NET Core Web Application (MVC).

Een GUI voor receptionisten en monteurs. Daarnaast wordt de communicatie met de verschillende services binnen deze appliatie geregeld.

### Onderdelen

De GA.FE bestaat uit de volgende onderdelen:

* Facade
* Infrastructure

#### Facade

De Facade bevat controllers, viewmodels en views. De Facade krijgt input van gebruikers en stuurt waar nodig gegevens terug.

Dit project heeft een referentie naar het Infrastructure project. De Infrastructure wordt hieronder besproken.

#### Infrastructure

De Infrastructure regelt de communicatie tussen de GA.FE en andere services.

De Infrastructure communiceert met zowel Services, als Integration Services. Op dit moment gaat het om twee servies waarmee wordt gecommuniceerd. De Case2.MaRoWo.OnderhoudBeheer.Service en de Case2.MaRoWo.RDW.IntegrationService. Communicatie wordt geregeld door middel van gegenereerde Agents. Deze Agents zijn opgebouwd met behulp van een swagger.json bestand. Deze zijn te verkrijgen van de services. Vervolgens wordt de swagger.json gebruikt met behulp van AutoRest om de Agents daadwerkelijk te genereren.

Het Infrastructure project is in deze solution de enige laag die met de ‘MaRoWoGarageAdministratie’ database communiceert. De solution beschikt over één database, dit is de ‘MaRoWoGarageAdministratie’ database. Het Infrastructure project beschikt over repositories om daadwerkelijk met de database te kunnen communiceren.

## Case2.MaRoWo.OnderhoudBeheer.Service (OB.Service)

Een ASP.NET Core Web Application (Web API).

Wat de GUI beteft kan er gebruik worden gemaakt van SwaggerUI. Dit is de landingspagina van de applicatie. Hierop worden alle beschikbare endpoints met mogelijk input weergegeven.

Deze service werkt op een Event-Driven manier. Daarnaast wordt gebruik gemaakt van de Onion Architecture. De OB.Service beschikt over een dispatcher voor het opvangen van events van RabbitMQ.

### Onderdelen

De OB.Service bestaat uit de volgende onderdelen:

* Facade
* Infrastructure
* Domain
* Incoming

#### Facade

De Facade bevat Web API controllers. De Facade ontvangt commands van GA.FE. De controllers zijn voorzien van Swagger attributes. Op basis van deze attributes wordt de swagger.json gegenereerd en kan er gebruik worden gemaakt van SwaggerUI.

De implementatie van deze commands staat gedefinieerd. De Facade stuurt commands door naar het Infrastructure project.

Dit project heeft een referentie naar het Infrastructure project. De Infrastructure wordt hieronder besproken.

#### Infrastructure

De Infrastructure regelt de communicatie tussen de OB.Service en andere services.

De Infrastructure heeft een dispatcher, hiermee kunnen events van RabbitMQ worden ontvangen. Op dit moment wordt enkel geluisterd naar events van de RDW Integration Service. Enkel deze events zijn relevant voor de OB.Service.

Het Infrastructure project is in deze solution de enige laag die met de ‘MaRoWoOnderhoudsBeheer’ database communiceert. De solution beschikt over één database, dit is de ‘MaRoWoOnderhoudsBeheer’ database. Het Infrastructure project beschikt over repositories om daadwerkelijk met de database te kunnen communiceren.

#### Domain

Het Domain bevat de database entiteiten. Daarnaast beschikt het Domain over DomainServices. Bij deze DomainServices wordt een bijbehorende repository en eventpublisher geïnjecteerd.

De DomainServices bevatten methodes voor het afhandelen van commands. De data uit deze commands wordt eventueel opgeslagen in de database. Daarna wordt een event opgebouwd en deze door middel van de eventpublisher naar RabbitMQ gestuurd.

#### Incoming

Het Incoming project bevat alle ondersteunde inkomende commands voor de OnderhoudBeheer service.

### Inkomende commands

Deze service ontvangt commands van GA.FE. De volgende commands kunnen worden verwerkt:

* CreateOnderhoudCommand
* OnderhoudAfmeldenCommand
* UpdateOnderhoudCommand

### Inkomende events

Deze service ontvangt events van Case2.MaRoWo.IntegrationService. De volgende events kunnen worden verwerkt:

* AfkAfgemeldEvent

### Uitgaande events

Deze service verstuurt events naar RabbitMQ. Andere services luisteren via RabbitMQ om events van deze service binnen te krijgen. De volgende events worden verzonden vanaf de service:

* OnderhoudsopdrachtCreatedEvent
* OnderhoudsopdrachtUpdatedEvent

## Case2.MaRoWo.RDW.IntegrationService (RDW.IS)

Een ASP.NET Core Web Application (Web API)

### Architectuur

Deze integration service werkt op een zowel Event-Driven manier als een op een sequentiele manier. Hiermee wordt bedoelt dat deze service events gooit, maar ook direct een response kan geven aan gebruikers van deze integration service

Daarnaast wordt gebruik gemaakt van de Onion Architecture.

### Inkomende commands

Deze integration service ontvangt commands van Case2.MaRoWo.GarageAdministratie.FrontEnd (GA.FE). De volgende commands kunnen worden verwerkt:

* ApkKeuringsVerzoekCommand

### Uitgaande requests

Deze integration service stuurt requests naar een externe server. De RDW APK Server. Na sturen van een request naar de RDW APK Server komt er een response binnen. Deze wordt afgehandeld door de

### Uitgaande events

Deze service verstuurt events naar RabbitMQ. Andere services luisteren via RabbitMQ om events van deze service binnen te krijgen. De volgende events kunnen door de service worden verzonden:

* ApkAfgemeldEvent

# Development

### Requirements voor Development

* Windows 10
* Visual Studio 2015 Update 3 (.NET Core)
* Docker for Windows
  + C schijf geshared
  + Minimaal **4096MB RAM** voor Docker reserveren in de Docker instellingen
* Microsoft SQL SERVER 2016

### Opstarten

Wanneer de source code moet worden aangepast, kan dit op de volgende manier worden gedaan:

* Ga naar “99-shared\deliverables” folder
* Open “Startup-Development.ps1”

Dit script maakt alles gereed voor ontwikkeling. Maakt een nieuw Docker network aan, doet een dotnet restore –no-cache voor het oplossen van project.json errors. Daarnaast wordt Visual Studio met de juiste solutions opgestart.

# Testhandleiding

Testen kunnen vanuit Visual Studio worden gedraaid. Ga naar de Test Explorer en klik op “Run all” per solution.

Testen wordt gedaan met behulp van MoQ en handgeschreven Mocks.

Bij het testen van de Infrastructure moet rekening worden gehouden met het testen van de database communicatie, middels de repositories. Tijdens dit project zijn we begonnen met testen op een in-memory database, echter bleek dat dit geen solide keuze was. De in-memory database beschikt niet over validatie van de in te voeren objecten. Dit heeft ons doen beslissen over te stappen naar een Local database. Bijkomend voordeel is dat deze keuze meer lijkt op een reële situatie.

# Troubleshooting

Het gebruik van .NET Core wil zo nu en dan problemen opleveren. Dit zien we voornamelijk terug wanneer een solution wordt geopend. In Error List komen dan de volgende errors voor: “An item with the same key has already been added.”.

Over het algemeen kan dit worden opgelost met het volgende command “dotnet restore –no-cache”. Om fouten te voorkomen, kunnen de Solutions het best worden geopend met het “Startup-Development.ps1” in de 99-shared\deliverables folder van deze repository.   
Dit voert voor alle solutions direct een **dotnet restore –no-cache** uit.