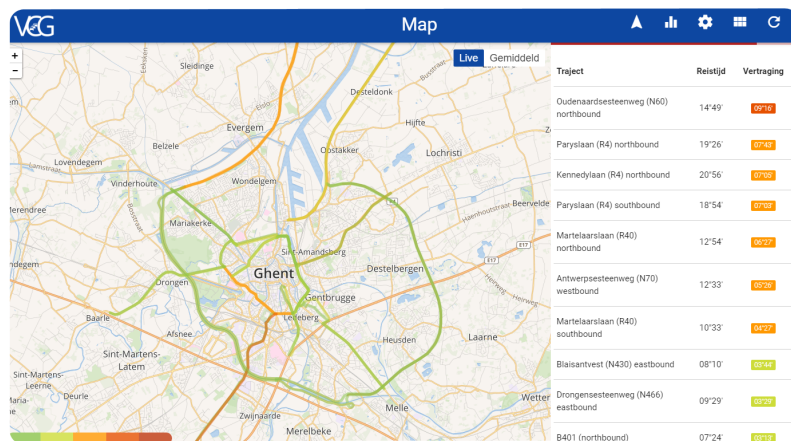




3e Bachelor Informatica  
Academiejaar 2015-2016

Faculteit Ingenieurswetenschappen en Architectuur  
Valentin Vaerwijckweg 1 - 9000 Gent

# Verkeerscentrum Gent



Verslag voor bachelorproef

Groep 2 Mike BRANTS  
Thomas VANDE WEGHE  
Tobias VAN DER PULST  
Simon VERMEERSCH

# Woord vooraf

---

Voor u ligt het eindverslag van onze bachelorproef 'Verkeerscentrum Gent'. Als studenten Industrieel Ingenieur aan Universiteit Gent hebben we veel geleerd door onze theorie naar praktijk om te zetten. Hierbij willen we alle mensen die ons met raad en daad hebben bijgestaan bedanken.

In het bijzonder bedanken we onze opdrachtgever Pieter Morlion van Mobiliteitsbedrijf Gent en zijn assistent Florian Goeteyn voor hun goede, vlotte samenwerking. We willen ook alle begeleiders bedanken, met een speciale vermelding voor Helga Naessens. Verder willen we ook onze ouders, familie en vrienden bedanken voor de steun.

# Abstract

---

## English

A lot of big cities all over the world have access to a database with traffic data and a modern traffic center. The city of Ghent does not meet current standards, which directly concerns the inhabitants because they have a lack of information. At the moment it's difficult to analyse what influence certain traffic changes have to the general traffic in the city. If the database is filled with data over multiple years, it will be possible to find the average traffic situation on a route.

The target of this thesis is to design an application that will collect structured data for a number of routes using traffic data sources. After a few years there will be a huge database which can be used to do numerous analyses. Thus, it will become clear which data source is fastest in detecting traffic changes and which routes are the most susceptible for delays. By using this information, an employee in the traffic centre will be able to draw conclusions and make changes to the current traffic plan.

The main target of the application is to inform the inhabitants of Ghent with better realtime information concerning changing traffic situations and traffic delays in the city. This way people can make decisions about which routes they are going to use and about the moment at which they will take the trip.

## Nederlands

Vele grote steden over de hele wereld beschikken over databases met gegevens uit het verkeer en een modern verkeerscentrum. De stad Gent loopt op dat vlak achter, wat jammer is voor de inwoners die op deze manier een gebrek aan informatie hebben. Op dit moment is het moeilijk te analyseren wat voor invloed bepaalde verkeerswijzigingen hebben op het verkeer in de hele stad. Indien er over meerdere jaren data wordt opgehaald per traject in de stad, dan zal men een goed idee hebben van de gemiddelde verkeerssituatie op een traject.

Het doel van de bachelorproef is een applicatie te ontwerpen die op basis van verschillende verkeersdatabronnen gestructureerde gegevens zal verzamelen voor een aantal trajecten in de stad Gent. Op deze manier zal op lange termijn een grote databank ontstaan aan de hand waarvan uiteenlopende analyses kunnen worden uitgevoerd. Zo zal blijken welke verkeersdatabron meestal als eerste reageert op vertraging, welke trajecten het gevoeligst zijn voor vertraging en dergelijke meer. Met behulp van deze informatie zal een werknemer in het verkeerscentrum de nodige conclusies kunnen trekken en eventuele wijzigingen doorvoeren aan het bestaande verkeersplan.

Het hoofddoel van de applicatie is om de inwoners van de stad Gent betere realtime informatie te geven over wijzigende verkeerssituaties en vertragingen in de stad. Op die manier kan men op individuele basis het beste traject en het beste moment kiezen om een verplaatsing te maken in de regio.

# Inhoudsopgave

---

<b>1</b>	<b>Behoeftanalyse</b>	<b>1</b>
1.1	Beschrijving project . . . . .	1
1.2	Functionaliteiten . . . . .	1
1.2.1	Basis . . . . .	1
1.2.2	Extra . . . . .	1
1.3	Use Case . . . . .	2
1.4	Functieanalyse van de omgeving . . . . .	6
<b>2</b>	<b>Ontwerp</b>	<b>7</b>
2.1	Functioneel ontwerp . . . . .	7
2.2	Technisch ontwerp . . . . .	7
2.2.1	Hardware . . . . .	7
2.2.2	Software . . . . .	7
2.2.3	Pakketten . . . . .	8
2.3	Software ontwerp . . . . .	12
2.3.1	Databronnen . . . . .	12
2.3.2	Verantwoordelijkheid per bean/map . . . . .	14
2.3.3	Verkeerlib . . . . .	15
2.3.4	Kernlaag . . . . .	18
2.3.5	Communicatielaag . . . . .	20
2.3.6	Presentatielaag . . . . .	21
2.3.7	Properties . . . . .	25
2.3.8	DataProvider - Methodes . . . . .	30
2.3.9	Databank . . . . .	32
2.3.10	Klassendiagram . . . . .	33
2.3.11	Gegevensstroomdiagram . . . . .	36
2.3.12	Line Of Business . . . . .	37
2.4	Data ontwerp . . . . .	38
2.4.1	Routes . . . . .	38
2.4.2	Analyses . . . . .	43
2.4.3	Providers . . . . .	53
2.4.4	API Keys . . . . .	54
2.4.5	Gebruikers . . . . .	56
2.4.6	Thresholds . . . . .	59
2.4.7	Timer . . . . .	62
2.4.8	Loggings . . . . .	64
2.4.9	GeoJson . . . . .	65
2.4.10	Instellingen . . . . .	67
<b>3</b>	<b>Kwaliteitscontrole</b>	<b>74</b>
3.1	Core . . . . .	74
3.1.1	Sortering GeoLocations . . . . .	74
3.1.2	GeoLocation . . . . .	74
3.1.3	GeoJsonProvider . . . . .	74

3.1.4	SourceAdapter	74
3.1.5	TimerScheduler	75
3.1.6	TrafficDataDownloader - DownstreamAnalyser - DAO	75
3.2	DataProvider - RESTAPI	77
3.3	Website	77
3.3.1	Startpagina	77
3.3.2	Live	77
3.3.3	Analyse	77
3.3.4	Instellingen	77
<b>4</b>	<b>Gebruikshandleiding</b>	<b>78</b>
4.1	Website	78
4.1.1	Live-pagina	79
4.1.2	Analyse-pagina	80
4.1.3	Instellingen-pagina	86
4.1.4	API	87
4.2	API	88
<b>5</b>	<b>Installatiehandleiding</b>	<b>89</b>
5.1	Systeemeisen	89
5.1.1	Systeemvereisten	89
5.1.2	Software	89
5.1.3	Software	89
5.2	Instellen systeem	90
5.2.1	Inladen modules	91
5.3	Beheer	92
5.3.1	Start/Stop	92
5.3.2	Logs bekijken	92
5.3.3	Aanpassen systeem	92
5.4	<i>Fixes</i> in Glassfish	93
5.5	Externe <i>libraries</i> in Glassfish	93
5.6	Glassfish Resources	94
5.7	Glassfish JDBC	96
<b>6</b>	<b>Conclusie</b>	<b>97</b>
<b>7</b>	<b>Taakverdeling</b>	<b>98</b>
7.1	Mike Brants	98
7.2	Thomas Vande Weghe	98
7.3	Tobias Van der Pulst	98
7.4	Simon Vermeersch	98
<b>8</b>	<b>Bijlage</b>	<b>99</b>
8.1	Testplan Website	99
8.1.1	Startpagina	99
8.1.2	Live	99
8.1.3	Analyse	101
8.1.4	Instellingen	107
<b>9</b>	<b>Referentielijst</b>	<b>109</b>

# Hoofdstuk 1: Behoeftanalyse

---

Het doel van dit project is het opzetten van een database voor de verkeersgegevens van bepaalde trajecten in Gent. In de eerste plaats worden gegevens opgehaald van bekende verkeersinformatie-databronnen. Daarna zal de opgehaalde data in een database opgeslagen worden. De bedoeling is om op deze manier de informatie van verschillende databronnen kwalitatief met elkaar te vergelijken. De data zal ook gebruikt worden om verkeerssituaties op verschillende routes te analyseren.

## 1.1 BESCHRIJVING PROJECT

---

Het Mobiliteitsbedrijf van de stad Gent is sinds 2014 bezig met het opzetten van een regionaal verkeerscentrum. Het is de bedoeling dat op termijn het verkeer in de regio constant gemonitord wordt, op semi-automatische basis op normale werkdagen en bemand tijdens piekmomenten en evenementen. Tijdens de week is het de bedoeling dat onverwachte incidenten, calamiteiten of significante verhogingen van de reistijden automatisch gesignaleerd worden aan de verantwoordelijke, die dan de nodige acties kan ondernemen. De gegevens zouden ook constant beschikbaar moeten zijn voor het publiek via een website, sociale media en open data. Op die manier kunnen mensen de beste route en het beste moment kiezen om een verplaatsing te maken in de regio.

## 1.2 FUNCTIONALITEITEN

---

### 1.2.1 Basis

1. Ophalen van kwalitatieve en vergelijkbare data bij verschillende bronnen
2. Opgehaalde data opslaan in database

### 1.2.2 Extra

1. Real-time overzicht van de verkeersdrukke op vooraf vastgelegde trajecten
2. Analyse en kwaliteitscontrole op basis van opgehaalde data
3. Aanbieden van gegevens aan externen via REST API
4. Platform gelinkt met sociale media om snelle communicatie aan te bieden
5. Meldingen genereren via sociale media (Twitter) wanneer reistijden overschreden worden
6. Bepalen van de oorzaak van een vertraging

## 1.3 USE CASE

### Algemeen

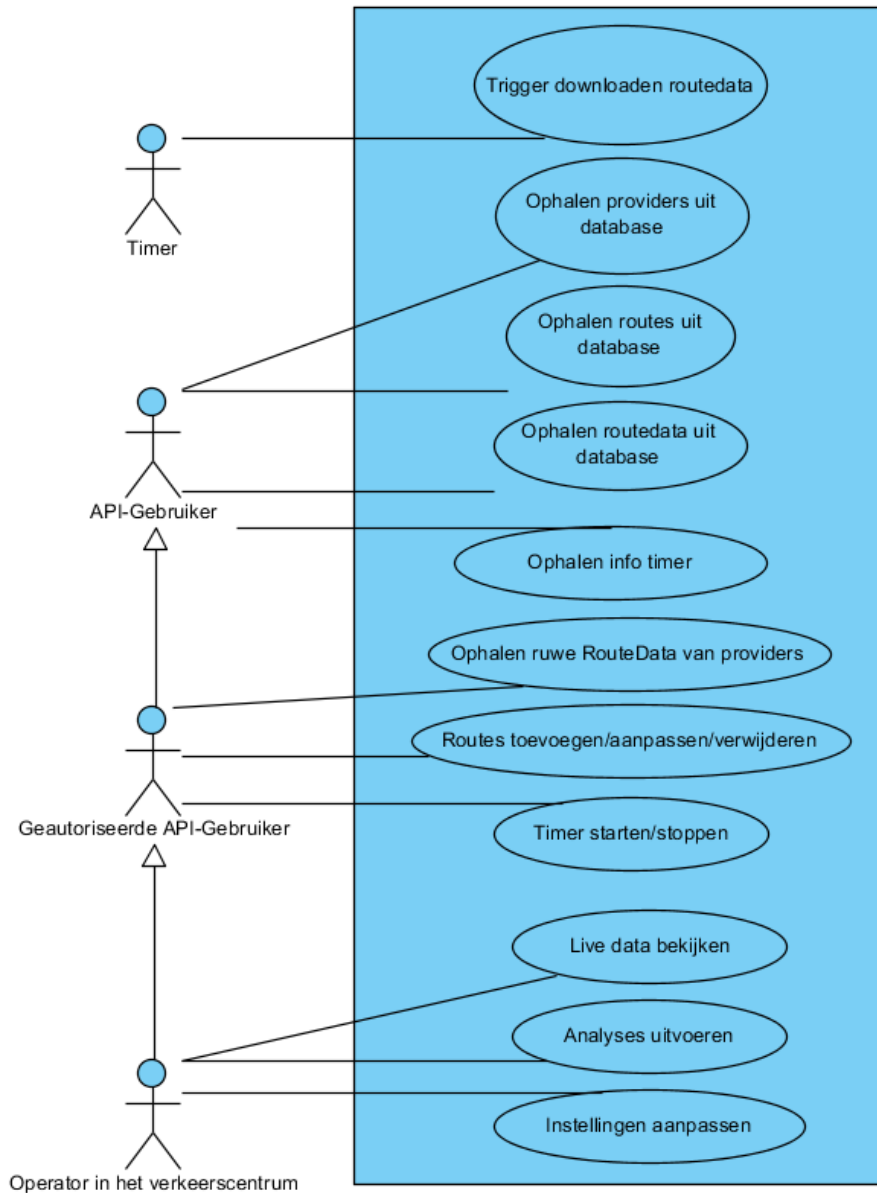


Diagram 1 Algemeen

In het gehele systeem zijn er drie mogelijke actoren. In de eerste plaats is er de Timer, deze zal een trigger sturen naar het programma zodat data afkomstig van de verschillende providers opgehaald wordt. Verder is er nog een API-gebruiker, dit is een persoon die data kan ophalen uit de database gebruikmakend van de RESTAPI. De API-gebruiker met een API-sleutel kan de data per provider ophalen. Een andere actor is de operator in het verkeerscentrum, hij zal via het dashboard de verkeersdata monitoren en de mogelijkheid hebben om grafische analyses uit te voeren. De operator zal met behulp van de website ook extra routes kunnen toevoegen waarvan data opgehaald moet worden.

## Website

In dit onderdeel worden de use cases van de website behandeld, hier is de gebruiker dus een operator in het verkeerscentrum. Deze persoon zal zich moeten aanmelden op de website voordat hij verdere stappen kan ondernemen.

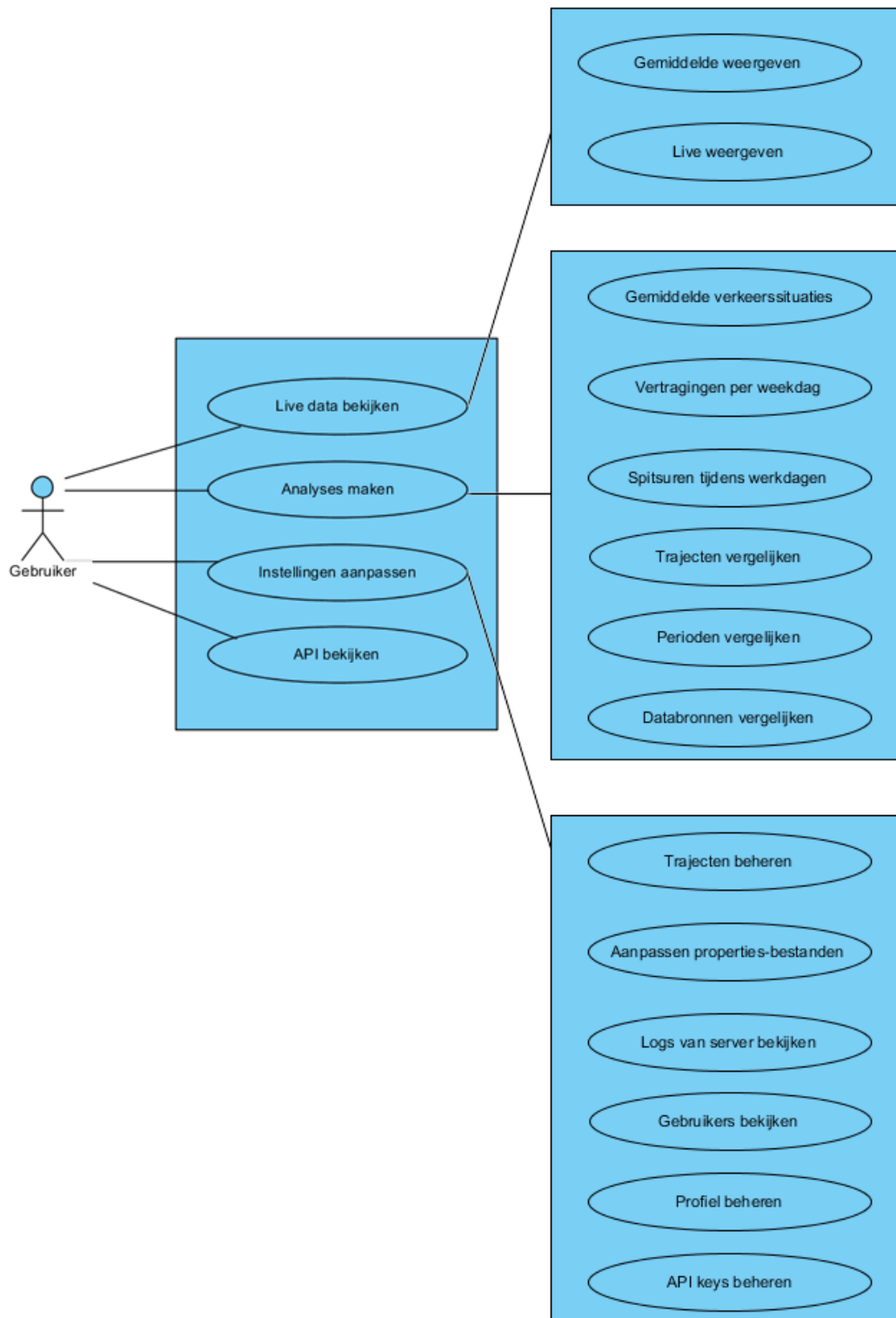


Diagram 2 Website (Algemeen)



Op de live-pagina is er meer informatie beschikbaar over de live-situatie, de gebruiker heeft ook de mogelijkheid om de gemiddelde situatie in plaats van de live-situatie te bekijken.

Op de website zijn er verschillende soorten analyses mogelijk, de meesten verlopen analoog en worden beschreven in diagram 3. In de eerste plaats is dit allemaal instelbaar via het verplichte stappenplan. In de analysepagina zelf heb je via de *sidebar* dezelfde keuzemogelijkheden waardoor je de aanvraag dynamisch kan wijzigen. De mogelijke analyses zijn: 'Gemiddelde verkeerssituaties', 'Vertragingen per weekdag' en 'Spitsuren tijdens werkdagen'. Elk van deze analyses heeft een specifieke uitkomst die verderop in de handleiding besproken wordt.

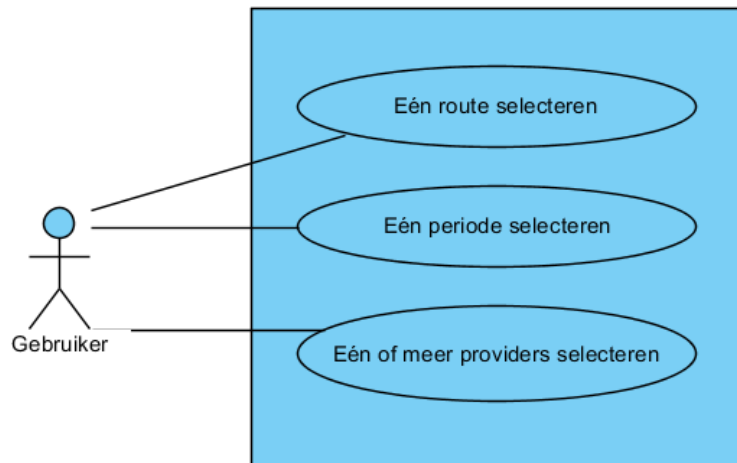


Diagram 3 Algemene analyse

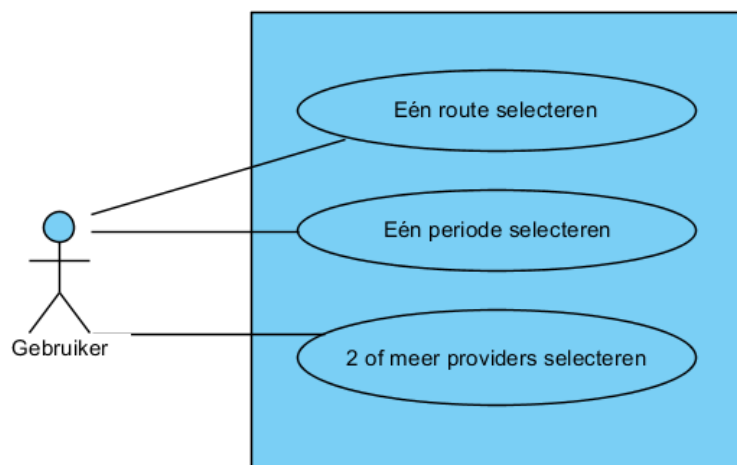


Diagram 4 Databronnen vergelijken

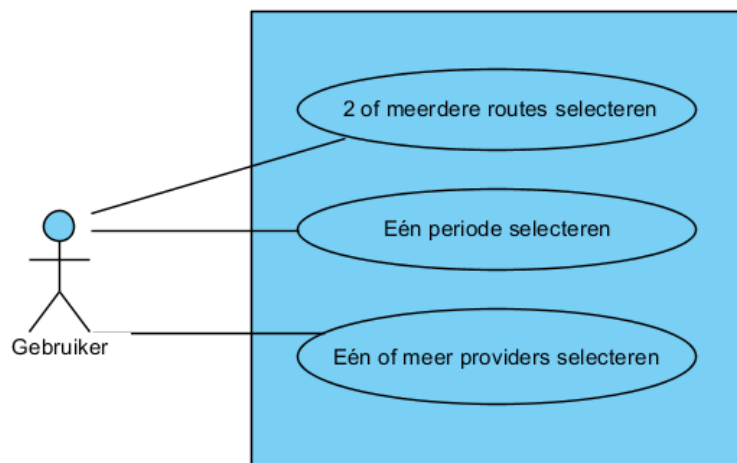


Diagram 5 Routes vergelijken

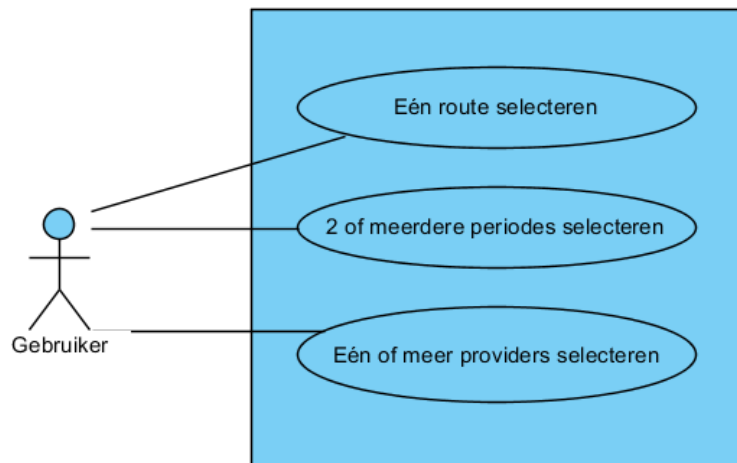


Diagram 6 Periodes vergelijken

## 1.4 FUNCTIEANALYSE VAN DE OMGEVING

---

### 1. Gebruikers

- (a) Ontwikkelaar
- (b) REST-gebruiker
- (c) Operator in het verkeerscentrum

### 2. Doelstellingen

De doelstellingen representeren de product backlog en ze bevatten de taken die het systeem moet kunnen.

#### (a) Basisfunctionaliteit

- i. Data ophalen uit meerdere bronnen
  - A. Google Maps
  - B. Here
  - C. Waze
  - D. TomTom
  - E. Coyote
- ii. Databank creëren en opvullen met opgehaalde data

#### (b) API met verschillende parameters

- i. Periode
- ii. Traject
- iii. Databron
- iv. Vertraging

#### (c) Dashboard voor analyse van de verkeerssituaties

- i. Grafische opbouw van de GUI
- ii. Grafieken/Tabellen genereren
- iii. Grafische weergave op kaart
- iv. Ophalen data aan de hand van API
- v. Kwaliteitscontrole van de verschillende databronnen

# Hoofdstuk 2: Ontwerp

---

## 2.1 FUNCTIONEEL ONTWERP

---

Er zijn twee types fysieke gebruikers in het systeem. Enerzijds zijn er de API-gebruikers, zij hebben de mogelijkheid om data uit de API op te vragen en eventueel verder te verwerken. Anderzijds zijn er de operatoren in het verkeerscentrum, zij hebben de mogelijkheid om via de webapplicatie live data te monitoren en analyses uit te voeren. Verder kan de operator ook nog routes toevoegen aan het systeem.

## 2.2 TECHNISCH ONTWERP

---

### 2.2.1 Hardware

De gevraagde applicatie is geprogrammeerd in Java, dit laat toe om op alle besturingssystemen te draaien zolang deze Java ondersteunen. Er zijn echter ook een aantal eisen aan de hardware voor deze applicatie. Enkele softwarepakketten zijn vereist voor het draaien van deze applicatie. Er is steeds een actieve internetverbinding vereist om de data van de databronnen te ontvangen.

#### Systeemvereisten

1. 1GB memory minimum (2GB memory recommended)
2. 500MB disk space minimum (1GB disk space recommended)
3. Supported platforms: Solaris 10, OpenSolaris 2009, Red hat enterprise linux 4+, Ubuntu linux 8+, Windows xp SP3+, Mac OS X 10.5+
4. JDK 7 minimum

### 2.2.2 Software

1. GlassFish Server Open Source Edition 4.1 (build 13) - Java ee container voor de werkelijke applicatie
2. MySQL server (Project tested on Linux Ubuntu with '5.5.47-0ubuntu0.14.04.1 (Ubuntu mariadb' ) - Database voor opslag data
3. (optioneel) *nginx* - *Instellen van forwarding naar http / https poorten van de bovenstaande applicaties*

### 2.2.3 Pakketten

Dit project wordt uitgewerkt in Java met behulp van het Java EE (Enterprise Edition) *framework*. Dit *framework* omvat verschillende technologieën die worden gebruikt in deze applicatie.

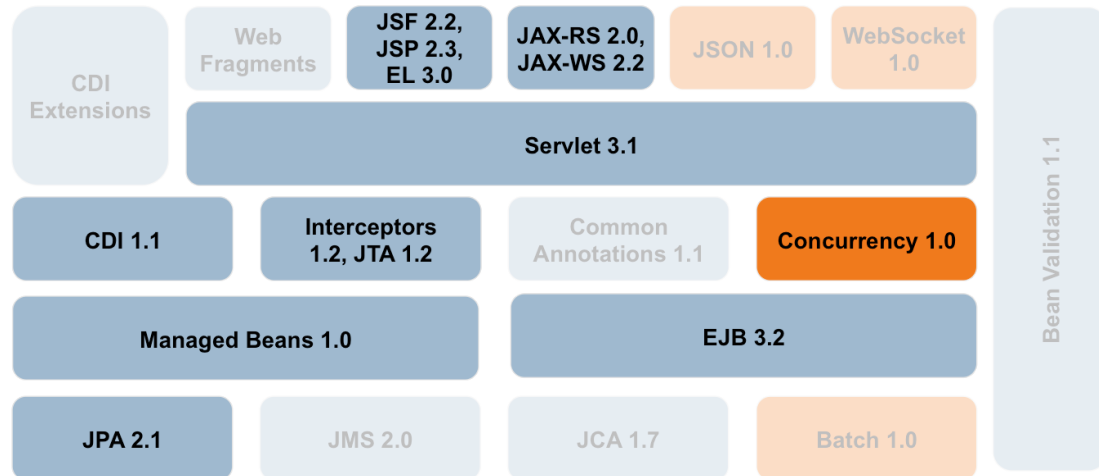


Diagram 7 Java EE

#### Kern

1. Context and Dependency Injection for Java 1.1
2. Enterprise JavaBeans (EJB) 3.2
3. Java Persistence API (JPA) 2.1
4. Java Transaction API (JTA) 1.2
5. Java API for RESTful Web Services (JAX-RS) 2.0
6. Managed Beans
7. Concurrency

Dit wordt aangevuld met Glassfish voorzieningen.

1. Java Naming and Directory Interface (JNDI)
2. Java Database Connection (JDBC) connection pools

#### Website

Er zijn ook nog een aantal pakketten die toegevoegd werden specifiek om de website te ondersteunen.

1. Expression Language (EL) 3.0
2. JavaServer Faces (JSF) 2.2
3. Sass
4. Servlet 3.1

Op de volgende pagina's zal elk deel uitgebreider behandeld worden.

## Java EE: Algemeen

Java EE laat 2 soorten interfaces tussen EJB's toe: Remote en Local. In deze applicatie zijn beide voorzien en actief. Wel worden over het gehele project de lokale beans gebruikt om zuiniger om te springen met het beschikbare geheugen van de testomgeving.

Een lokale interface geeft aan dat de bean zich binnen dezelfde Java Virtual Machine bevindt (op dezelfde server, binnen hetzelfde proces). Zo kunnen de beans rechtstreeks met elkaar communiceren, hun functies gebruiken en objecten met elkaar delen via het gemeenschappelijk geheugen. Dit is hetzelfde gedrag dat een gewone Java klasse vertoont ten opzichte van een andere Java klasse.

Een Remote-interface daarentegen geeft aan dat de andere bean zich mogelijk niet op dezelfde server bevindt. Men zal alle objecten moeten serialiseren en afleveren aan de container, omdat men geen objecten kan delen via het geheugen. Zo zal de andere server aangesproken worden en de data worden overgebracht. Hierbij is het zeer belangrijk dat alle objecten worden doorgegeven het '*pass-by-value*'-principe. Er is geen terugkoppeling naar de originele bean naast de returnwaarde van de functie.

Hoewel de Remote-interface ook lokaal werkt en dus een hogere functionaliteit vertoont t.o.v. de lokale interfaces gaat dit wel gepaard met een enorme toename aan geheugengebruik. Dit ontstaat door het vele serialisatiewerk van de beans en hun functies met parameters.

## Contexts and Dependency Injection

Alle beans verwachten zekere diensten waarvan zijzelf ook afhankelijk zijn. Zo wordt bij de Data Access Beans (xxxDAO.java) gebruik gemaakt van de EntityManager voor interactie met de databank. Bij de TrafficDataDownloader wordt dan weer de context van de applicatieserver verwacht om andere beans op te vragen. Al deze diensten zijn niet de verantwoordelijkheid van deze individuele beans maar van de applicatieserver. Deze laatste gedraagt zich als *injector* en zal alle vereiste *services* (*dependencies*) injecteren in de beans aan de hand van annotaties en objecttypes.

## EJB

JavaBeans zijn door software beheerde modulaire bouwblokken. In deze beans wordt de *business logic* voor een Enterprise Applicatie verwerkt. De grootste kenmerken van deze beans zijn hun modulariteit, onafhankelijkheid van elkaar en schaalbaarheid.

### *Modulariteit*

Iedere module (EJB) in het project is uitwisselbaar met een nieuwe module. Dit is aan te passen in een extern *properties*-bestand. Zo kan op ieder moment bijvoorbeeld een databron worden toegevoegd, een database worden vervangen door een andere of een nieuwe *web service* worden toegevoegd.

### *Onafhankelijkheid*

Alle beans zijn onafhankelijk. Alle objecten die voorkomen in meerdere beans (zoals interfaces) zijn gebundeld in een gemeenschappelijke bibliotheek. Hierdoor zullen beans niets merken wanneer een andere bean wijzigt.

### *Uitbreidbaarheid*

De modules zijn niet enkel onafhankelijk, maar hebben eveneens geen vaste relatie met de locatie waar ze werken. Zo kan een *database-bean* op een andere server staan dan de *analyser-bean*. De enige vereiste hiervoor is dat JNDI van de ene server gelinkt is aan de JNDI van de andere server. De beans zullen hun parameters en teruggeefwaardes steeds serialiseren en doorsturen naar de zogenaamde *remote bean*.

## JPA

De Java variant voor Object Relational Mapping (ORM) laat toe de gegevens in een databank rechtstreeks af te beelden op objecten door middel van annotaties. Deze manier van interactie met de databank laat een zeer eenvoudige werking toe. Het zal echter niet de performantie van handmatige SQL-commando's evenaren.

## JTA

Deze API start (zonder enige configuratie) steeds een transactie bij het aanroepen van een functie in een *managed bean*. Indien die functie een fout zou opwerpen zal een *rollback* gebeuren tot de toestand vlak voor de aanroep is bereikt. In deze applicatie wordt op deze API vertrouwd voor opslag van gegevens in de databank. Bij een *error* zal de opdracht voor dat interval niet worden uitgevoerd, maar zal de applicatie wel blijven werken.

## JAX-RS

Deze API laat toe om services aan te bieden volgens het Representational State Transfer (REST) patroon. In deze applicatie wordt het gebruikt om de API uit te werken.

## JNDI

Deze technologie laat toe data of objecten op te vragen via hun naam. Voor dit project werd de link naar de bronbestanden, de link naar de JDBC Connection Pool en de link naar alle beans opgenomen in JNDI.

## JDBC Connection Pools

Een Connection Pool houdt een *cache* van connecties naar een welbepaalde databank bij en maakt deze beschikbaar aan de applicaties van de applicatieserver. Dit alles zorgt voor een hogere efficiëntie want de connecties worden behouden en herbruikt. Hiernaast wordt er ook een hogere veiligheid aangeboden, de connectieparameters zijn namelijk niet langer in de applicatie zelf aanwezig.

## Managed Beans

Managed Beans worden beheerd door het JSF Framework. Zo laten ze toe om andere beans te gebruiken via '*injection*'. Beans kunnen ook gebruikt worden door de EL-API voor het weergeven van data op een gegenereerde html-pagina. Via annotaties kan je eveneens hun werkingsgebied definiëren. Dit werkingsgebied kan gedeeld worden over de hele applicatie (*ApplicationScoped*), binnen een sessie (*SessionScoped*) of zelfs binnen een aanvraag (*RequestScoped*) zijn.

## Concurrency

De Concurrency-API breidt de standaard Java '*java.util.concurrent*'-*package* uit. Hierbij maakt de module gebruik van de JNDI-Tree om het beheer van de *concurrent* objecten door te geven naar de *container* in plaats van de gebruiker. Dit is een extra ingebouwde veiligheid waarbij de applicatie zelf waakt of *concurrency* steeds wordt gerespecteerd.

## EL

Expression Language (ook aangeduid als EL) is een belangrijk mechanisme die de presentatielaag (webpagina's) laat communiceren met de toepassingslogica (Managed Beans). De EL wordt gebruikt door zowel de JavaServer Faces-technologie en JavaServer Pages-technologie.

## JSF

Deze technologie laat toe webpagina's te genereren op basis van data binnen de Managed Beans. Over de jaren heen is JSF uitgebreid met AJAX-ondersteuning, alsook *listeners* bij datawijzigingen. Dit alles laat toe een zeer flexibele, responsieve interface te maken voor de gebruikers.

## SASS

SASS is een framework dat een handige uitbreiding biedt aan CSS: genestelde regels, variabelen, functies, overerving, mixins, selectors, en veel meer. Het vertaalt SCSS (Sassy-CSS) naar welgevormde, standaard CSS gebruik makend van de commandoprompt of via plugins.

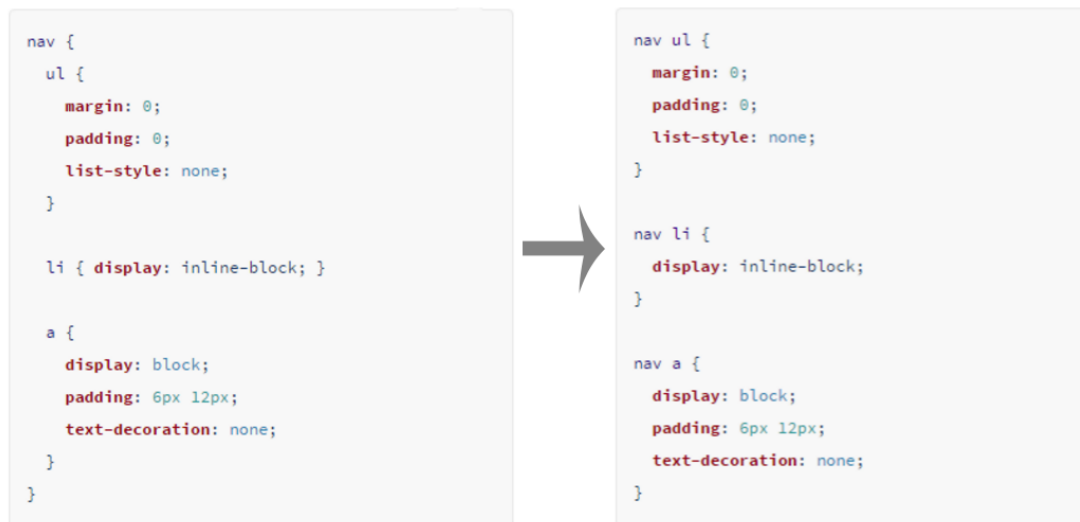


Diagram 8 Voorbeeld SASS

## Servlet 3.1

Servlets zorgen voor een componentgebaseerde, platformonafhankelijke methode voor het bouwen van webgebaseerde applicaties, zonder beperkingen van CGI-programma's. Servlets hebben toegang tot de hele familie van Java API's, inclusief de JDBC-API om toegang te krijgen tot databases.



## 2.3 SOFTWARE ONTWERP

---

### 2.3.1 Databronnen

#### Google Maps

*URL: <https://developers.google.com/maps/documentation/distance-matrix/>*

Reistijden van Google Maps kunnen opgevraagd worden via de Google Maps Distance Matrix API. In de URL kunnen verschillende start- en eindpunten worden meegegeven. Er moet rekening gehouden worden dat er voor elke combinatie van start- en eindpunt een reistijd wordt opgenomen in het antwoord. Bij het opgeven van drie startpunten en drie eindpunten, zal het resultaat een 3X3-matrix zijn. Dit komt overeen met negen aanvragen. Er kunnen maximaal 10 start- en eindpunten worden opgegeven in één aanvraag.

Om aanvragen te doen naar de API, is er een unieke toegangssleutel nodig, die kan aangevraagd worden bij Google. Met de gratis sleutel zijn er 2500 aanvragen per dag mogelijk. Indien deze limiet overschreden is, wordt er 0,50 dollar (= 0,4548 euro) per 1000 extra elementen aangerekend. In de huidige situatie waarbij er van 34 routes elke vijf minuten data wordt opgevraagd en dit 18u per dag, is er nood aan 7300 aanvragen per dag. Indien er rekening gehouden wordt met extra tussenpunten zal het aantal aanvragen sterk oplopen. Het gratis model van Google zal niet volstaan. Er kan worden overgeschakeld op het 'Google Maps API's Premium Plan' zodat er tot 100 000 aanvragen per dag gedaan kunnen worden.

#### Here

*URL: <https://developer.here.com/rest-apis/documentation/routing>*

Here stelt reistijden ter beschikking via zijn Routing API. In de URL kan je een route meegeven door de coördinaten in te stellen voor start- en eindpunt. Ook eventuele tussenpunten kunnen worden vermeld. Verder moet voor de toepassing die hier ontworpen wordt steeds aangegeven worden dat men de kortste route wil en dat men informatie baseert op het huidige verkeer. Op deze manier zal steeds actuele verkeersinformatie over een vaste route worden teruggeven.

Om de API van Here te kunnen gebruiken moeten er twee sleutels aangevraagd worden. De eerste 90 dagen kan dit gratis en mogen er tot 100 000 aanvragen per maand gedaan worden. In de huidige situatie waarbij er voor 34 routes elke vijf minuten data wordt opgevraagd en dit 18u per dag, is er nood aan 220 000 aanvragen per maand. Dit betekent dat er sowieso een betalende formule nodig is. Om tot 275 000 aanvragen per maand te kunnen doen, moet er gekozen worden voor het standaardplan dat 99 euro per maand kost.

## TomTom

URL: <http://developer.tomtom.com/products/onlinenavigation/onlinerouting>

TomTom heeft een API waarbij je de verkeersinformatie van de routes kan opvragen waarna je een JSON-object terugkrijgt. Het gebruikte account is een evaluatieversie waarbij er een aantal beperkingen zijn. Eén van de beperkingen is dat er slechts vijf aanvragen per seconde kunnen gedaan worden en het is ook zo dat er op een hele dag in totaal slechts 1000 aanvragen mogen gedaan worden. Indien er van 34 routes data wordt opgehaald zit je al na drie uur over deze grens. Er zal bijgevolg sowieso moeten overgeschakeld worden naar een betalend plan, info daarover is niet te vinden op de site van TomTom. Er zal dus contact opgenomen moeten worden met de provider.

## Coyote

URL: <https://maps.coyotesystems.com/traffic/>

Coyote stelt geen API ter beschikking; alle data wordt verkregen door de website te *scrapen*. Er zijn bijgevolg geen beperkingen op het aantal aanvragen. De enige voorwaarde is dat je een account moet hebben op de website.

De applicatie van Coyote bevat een aantal routes waarvan de namen moeten gemapt worden naar de routenamen in onze applicatie. Hiervoor bestaat het CoyoteMapping.properties-bestand.

## Waze

URL: <https://www.waze.com/trafficview>

Waze verplicht je om routes voor te definiëren via hun website. De informatie over deze routes kan bekomen worden met behulp *AJAX-calls*. De gebruikte *AJAX-calls* werden bekomen door de broncode van hun website te analyseren. De ontvangen JSON bevat alle routes met info voor de ingelogde gebruiker.

De beveiliging van Waze is zeer beperkt, zo kan bovenstaande *AJAX-call* uitgevoerd worden zonder in te loggen op hun webservice. Zolang de juiste unieke id van de gebruiker wordt meegegeven kan de data worden opgevraagd.

Via een *properties*-bestand worden alle routes uit onze database vertaald naar hun Waze-variant (via opzoeking van *key-value pairs*).

De WazeSourceAdapter werkt via het *sessionID* principe zodat slechts één keer per *datascrub* (ophalen van data op een welbepaald tijdstip) alle data wordt binnengehaald. De TrafficData-Downloader genereert voor iedere *datascrub* een unieke, nieuwe *sessionID*.

### 2.3.2 Verantwoordelijkheid per bean/map

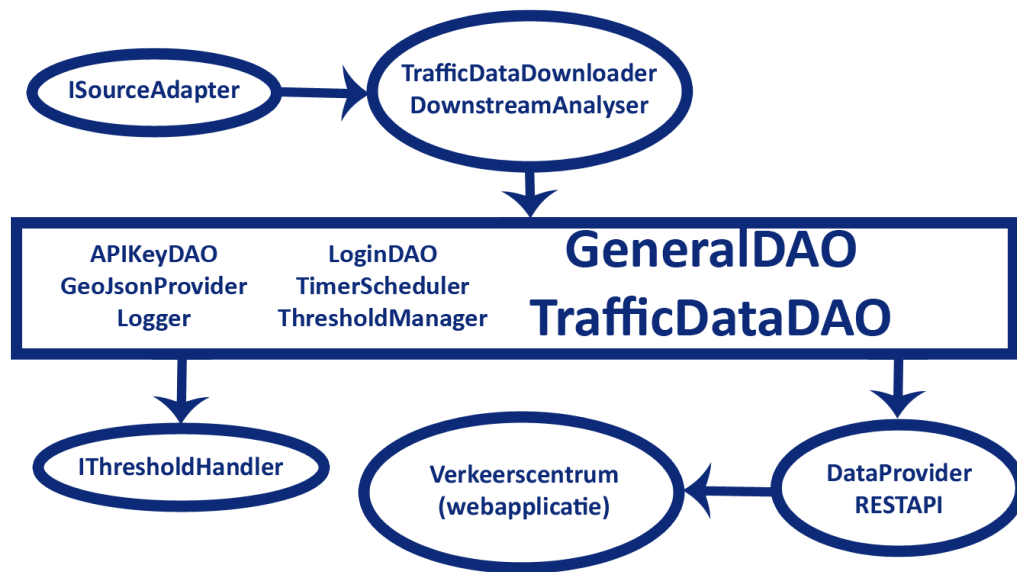


Diagram 9 Overzicht

Bean/Map	Verantwoordelijkheid/nut
Kernlaag	
APIKeyDAO	genereert en valideert keys voor gebruik API
[Databron]SourceAdapter	omzetten van data, aangeboden door een provider, naar RouteData-objecten voor één Route
GeneralDAO	zorgt ervoor dat de basiscomponenten (Routes, GeoLocations, Thresholds, ..) in de database terechtkomen
GeoJsonProvider	hulpbean om routes op kaart weer te geven
\lib	map waarin alle gebruikte <i>libraries</i> zitten
Logger	zorgt ervoor dat er gelogd kan worden en dat deze logs weggeschreven worden naar een bestand
LoginDAO	valideert login op website
Properties	beheer van het Properties.properties bestand
Thresholds	beheert de <i>thresholds</i> per Route
TimerScheduler	houdt alle info met betrekking tot de tijd en intervallen (wanneer de TrafficDataDownloader moet getriggerd worden etc.) bij
TrafficDataDAO	zorgt voor de opslag van RouteData in de database
TrafficDataDownloader	overlopen van alle Routes en zorgt ervoor dat per Route elke SourceAdapter RouteData ophaalt
TrafficDataDownstreamAnalyser	data afkomstig van SourceAdapters controleren op correctheid en nadien verdere acties ondernemen indien nodig
Verkeerlib	basiscomponenten en interfaces voor de beans
Communicatielaag	
DataProvider	haalt data uit de database aan de hand van bepaalde berekeningen en voorwaarden
RESTAPI	gebruikt data verkregen uit DataProvider en zet deze om naar JSON-objecten zodat API-aanvragen mogelijk zijn
Presentatielaag	
TwitterHandler	genereert berichten op Twitter indien aan bepaalde voorwaarden voldaan is
Verkeerscentrum	website van de applicatie

### 2.3.3 Verkeerlib

De Verkeerlib is de basiscomponent van de volledige applicatie. Deze library bevat elke interface die in de beans gebruikt kan worden. Indien een ontwikkelaar een bean wil toevoegen aan de applicatie dan zal deze persoon zich moeten baseren op de hieronder vermelde interfaces. Onderstaande tabel bevat alle informatie over de verschillende klassen en interfaces in de Verkeerlib, de tabel bestaat uit twee kolommen: de naam en de verantwoordelijkheid.

Merk onder andere op dat er een GeneralDAO en een TrafficDataDAO is, deze zijn gesplitst zodat op termijn bijvoorbeeld kan overgegaan worden naar een NoSQL-database voor de RouteData.

Verkeerlib	
APIKey	
APIKey	maakt het mogelijk API Keys te maken/beheren/verwijderen
Auth	
AuthHelpers	hulpklasse voor encryptering wachtwoorden
AuthUser	maakt het mogelijk gebruikers voor de website te maken/beheren/verwijderen
Components	
(I)GeoLocation	bevat informatie over een locatie (latitude en longitude)
GeoLocationComparator	vergelijkt GeoLocaties op hun volgorde in een bepaalde route
Log	maakt van logs objecten
IPeriod	bevat start- en eindpunt van een periode
(I)Route	bevat informatie over een route (id, naam en GeoLocations)
(I)RouteData	bevat opgehaalde data (afstand en reistijd) op een bepaalde tijdstip voor een bepaalde route
(I)Threshold	bevat een level, een delaylevel en een route, deze zullen bepalen hoeveel vertraging (delaylevel) nodig is om een bepaald (threshold)level te bereiken voor een route
Weekdays	enumeration die de namen van alle dagen (maandag tot en met zondag) bevat
DAO	
(I)APIKeyDAOLocal/Remote	API keys opslaan in de database
Aggregation	enumeration gebruikt voor communicatie met de TrafficDataDAO die de zoekopdracht van de DAO vastlegt, zo kan gevraagd worden de som van alle verkregen waarden terug te geven of dat de samenvoeging geen rekening moet houden met de datum (dus enkel tijdstip) van de data.
AggregationContainer	een wrapper voor Aggregation die tevens de gevraagde variabele bevat, Voorbeeld: new AggregationContainer(Aggregation.sum, "duration * distance") zal de som van alle 'durations * distances' teruggeven als return waarde.
(I)GeneralDAOLocal/Remote	Routes, GeoLocations(Mapping) en Thresholds opslaan in de database
(I>LoginDAOLocal/Remote	AuthUsers opslaan in de database
(I)TrafficDataDAOLocal/Remote	RouteData opslaan in de database

DataDownloader	
ISourceManager	bevat één methode (parse) die voor alle SourceAdapters RouteData zal ophalen bij een bepaalde Route
(I)TrafficDataDownloaderLocal/Remote	bevat één methode (downloadNewData) die getriggerd wordt door de Timer en die op een gegeven tijdstip RouteData zal ophalen voor alle Routes bij alle SourceAdapters (via de SourceManager)
DataProvider	
(I)DataProviderLocal/Remote	bevat verschillende methodes die op verschillende manieren data ophalen uit de database, deze staan verderop in dit document in detail besproken
DataSources	
(I)SourceAdapterLocal/Remote	bevat één methode (parse) die gebruikt wordt om RouteData van één databron op te halen voor een gegeven route
Downstream	
ITrafficDataDownstreamAnalyser	bevat een methode waarin eventueel bepaalde stappen kunnen ondernomen worden met RouteData vooraleer deze (al dan niet) in de database worden opgeslagen
TrafficDataDownstreamAnalyserLocal	<i>cfr. ITrafficDataDownstreamAnalyser</i>
TrafficDataDownstreamAnalyserRemote	<i>cfr. ITrafficDataDownstreamAnalyser</i>
GeoJson	
(I)GeoJsonLocal/Remote	hulpinterface om trajecten weer te geven op kaart
Helper	
BeanFactory	deze klasse bevat methodes om de verschillende beans op te vragen
BeanSelector	enumeration met de namen van de beans
DataAccessException	exceptieklasse, deze exceptie wordt opgegooid wanneer het niet mogelijk is om RouteData te verkrijgen
HelperFunctions	deze klasse bevat methodes die in BeanFactory worden gebruikt om verschillende beans op te vragen
InvalidCoordinateException	exceptieklasse, deze exceptie wordt opgegooid indien je een GeoLocation probeert te maken met onmogelijke coördinaten
NoInternetConnectionException	exceptieklasse, deze exceptie wordt opgegooid indien er geen internetconnectie is
ResourceFileMissingException	exceptieklasse, deze exceptie wordt opgegooid wanneer een bepaalde <i>resource</i> niet bereikbaar is in GlassFish
URLException	exceptieklasse, deze exceptie wordt opgegooid wanneer het niet mogelijk is om een opgegeven URL te bereiken
Logger	
(I)LoggerLocal/Remote	voorziet de applicatie van logging
Properties	
(I)PropertiesLocal/Remote	beheer standaarddatabronnen

Threshold	
(I)ThresholdManagerLocal/Remote	houdt de mapping van thresholdlevels per route bij
(I)ThresholdHandlerLocal/Remote	afhandelen van thresholds
Timer	
ITimer	houdt alle info met betrekking tot de tijd en intervallen (wanneer de TrafficDataDownloader moet getriggerd worden etc.) bij
TimerSchedulerLocal/Remote	<i>cfr. ITimer</i>

### 2.3.4 Kernlaag

Hieronder vindt u een oplijsting van alle bestaande beans die behoren tot de kernlaag met de klassen die zich in deze beans bevinden. Onderstaande tabellen bevatten alle informatie over de verschillende klassen en interfaces in de verschillende beans, de tabellen bestaan uit twee kolommen: de naam en de verantwoordelijkheid.

APIKeyDAO	
APIKeyDAO	genereert en valideert keys voor gebruik API
[Databron]SourceAdapter	
[Databron]SourceAdapter	omzetten van data, aangeboden door een bepaalde provider, naar RouteData voor één Route
[Databron]JUnitTest	unittest met betrekking tot de parse-methode in SourceAdapter
GeneralDAO	
GeneralDAO	zorgt ervoor dat de hieronder opgesomde klassen in de database terechtkomen
GeoLocationEntity	deze klasse vertegenwoordigt de GeoLocation-klasse zodat ze in de database kan opgeslagen worden
GeoLocationMappingEntity	deze klasse vertegenwoordigt de mapping van GeoLocations per Route zodat deze in de database kan opgeslagen worden
RouteEntity	deze klasse vertegenwoordigt de Route-klasse zodat ze in de database kan opgeslagen worden
ThresholdEntity	deze klasse vertegenwoordigt de Threshold-klasse zodat ze in de database kan opgeslagen worden
GeoJsonProvider	
GeoJsonProvider	hulpbean om routes op kaart weer te geven
GeoJsonProviderTest	een aantal unittests met betrekking tot de GeoJsonProvider
GeoJsonProviderTestBeanInfo	een aantal unittests met betrekking tot de GeoJsonProvider
Logger	
Logger	zorgt ervoor dat er gelogd kan worden en dat deze logs weggeschreven worden naar een bestand
LoginDAO	
LoginDAO	valideert login op website
Properties	
Properties	beheer standaarddatabronnen
Thresholds	
ThresholdManager	het beheer van de verschillende thresholds per route wordt hier voorzien

TimerScheduler	
TimerScheduler	automatische triggering voor ophalen nieuwe data

TrafficDataDAO	
Blocklist	opdelen van TrafficData-records in de database door middel van indexering aan de hand van timestamps
Operation	hulpklasse met enumeration van een aantal SQL-methodes
Parameter	hulpklasse om parameters toe te voegen aan een SQL-opdracht
Request	hulpklasse om SQL-opdrachten op te stellen
RouteDataEntity	deze klasse vertegenwoordigt de RouteData-klasse zodat ze in de database kan opgeslagen worden
TrafficDataDAO	zorgt voor de opslag van RouteData-objecten in de database

TrafficDataDownloader	
SourceManager	de verschillende SourceAdapters worden hier bijgehouden
TrafficDataDownloader	overlopen van alle Routes en zorgt ervoor dat per Route elke SourceAdapter (met hulp van de SourceManager-klasse) RouteData ophaalt

TrafficDataDownstreamAnalyser	
TrafficDataDownstreamAnalyser	data afkomstig van SourceAdapters controleren op correctheid en nadien verdere acties ondernemen indien nodig



### 2.3.5 Communicatielaag

Hieronder vindt u een oplistijng van alle bestaande beans die behoren tot de communicatielaag met de klassen die zich in deze beans bevinden. Onderstaande tabellen bevatten alle informatie over de verschillende klassen en interfaces in de verschillende beans, de tabellen bestaan uit twee kolommen: de naam en de verantwoordelijkheid.

DataProvider	
DataProvider	haalt data uit de database aan de hand van bepaalde berekeningen en voorwaarden

RESTAPI	
RESTAuthorizationFilter	beveiliging van de REST-aanvragen
CorsFilter	hulpklasse voor beveiliging van de REST-aanvragen
ApplicationConfig	klasse die bepaalt welke REST-klassen gebruikt worden
GeoJsonResource	alle REST-aanvragen met betrekking tot de GeoJsonProvider worden hier behandeld
Helper	hulpklasse voor de REST-aanvragen
LoggingsResource	alle REST-aanvragen met betrekking tot de serverlogs worden hier behandeld
ProviderResource	alle REST-aanvragen met betrekking tot de Providers/SourceAdapters worden hier behandeld
RoutesResource	alle REST-aanvragen met betrekking tot de Routes worden hier behandeld
SettingsResource	alle REST-aanvragen met betrekking tot de Users, APIKeys en properties-bestanden worden hier behandeld
TestResource	klasse gebruikt voor testing REST-aanvragen
ThresholdsResource	alle REST-aanvragen met betrekking tot de Thresholds worden hier behandeld
TimersResource	alle REST-aanvragen met betrekking tot de TimerScheduler worden hier behandeld

### 2.3.6 Presentatielaag

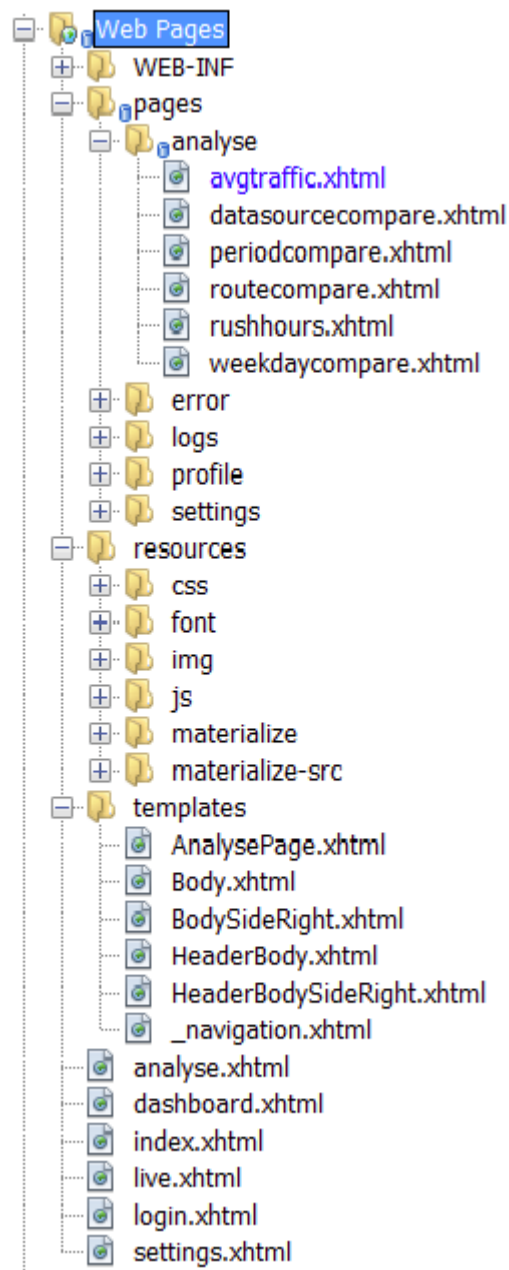
Hieronder vindt u een olijsting van alle bestaande beans die behoren tot de presentatielaag met de klassen die zich in deze beans bevinden. Onderstaande tabel bevat alle informatie over de verschillende klassen en interfaces in de TwitterHandler-bean, de tabel bestaat uit twee kolommen: de naam en de verantwoordelijkheid.

TwitterHandler	
Encoder	hulpklasse voor TwitterHandler
TwitterHandler	genereert berichten op Twitter indien aan bepaalde voorwaarden voldaan is

#### Verkeerscentrum (webapplicatie)

Het grootste deel van de presentatielaag bestaat uiteraard uit de webapplicatie.

De webapplicatie is gebouwd op het JSF framework. De Managed Beans die de backend van de webapplicatie verzorgen zijn terug te vinden in de source-map van het project. De bestanden voor de clientside zijn allen terug te vinden in de 'Web Pages'-map. Deze laatste map werd verder onderverdeeld om html, CSS, Javascript en afbeeldingen gescheiden te houden. De structuur wordt verduidelijkt in de afbeelding op de volgende pagina.



**Diagram 10 Algemene structuur**

De map 'resources' bevat alle CSS (+ Sassy-CSS bronbestanden), afbeeldingen en Javascript die worden gebruikt in de webpagina's. De map 'pages' zorgt voor een duidelijke structuur die de samenhang van bepaalde webpagina's benadrukt.

De backend werd onderverdeeld in verschillende *packages*: een *package* per module. De verschillende *packages* zijn terug te vinden in onderstaande afbeelding.

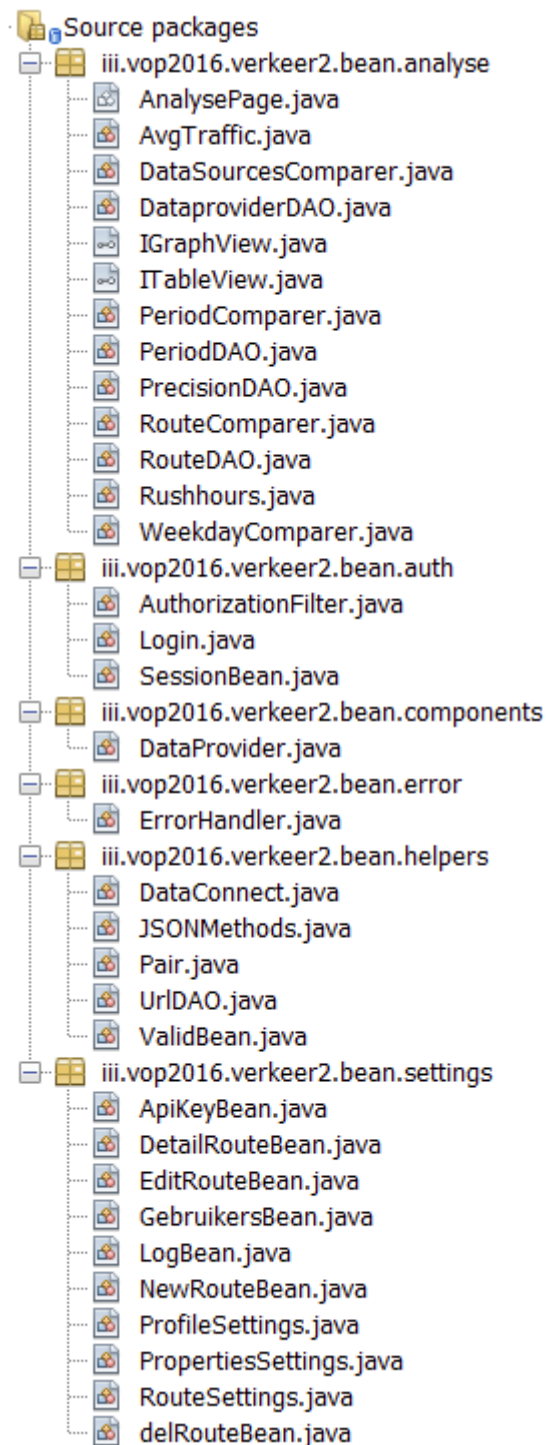
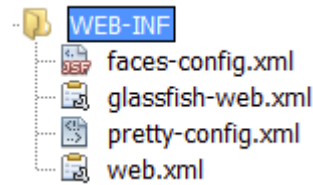


Diagram 11 Packages

In de volgende afbeelding zijn verschillende configuratiebestanden terug te vinden. 'faces-config.xml' bevat de configuratie voor de JSF-pagina's. Aangezien alles gebeurt via annotaties is dit bestand niet gevuld.



**Diagram 12 Configuratiebestanden**

'Pretty-config.xml' legt een mapping vast die url's afbeeldt op de desbetreffende webpagina. Een voorbeeld is te vinden in onderstaande afbeelding.

```
<url-mapping id="live">
    <pattern value="/live" />
    <view-id value="/faces/live.xhtml" />
</url-mapping>
```

**Diagram 13 Voorbeeld Pretty-config.xml**

## 2.3.7 Properties

### Beans

Het Beans.properties-bestand bevat verwijzingen naar de beans die opgevraagd kunnen worden in de BeanFactory. Indien er dus van een andere DownstreamAnalyser gebruik gemaakt moet worden zal in onderstaand voorbeeld de verwijzing moeten veranderen.

*Voorbeeld*

---

```
downstreamAnalyser=java:global/Verkeer2/TrafficDataDownstreamAnalyser/  
    TrafficDataDownstreamAnalyser!iii.vop2016.verkeer2.ejb.  
    downstream.TrafficDataDownstreamAnalyserLocal  
dataManager=java:global/Verkeer2/TrafficDataDownloader/TrafficDataDownloader!  
    iii.vop2016.verkeer2.ejb.datadownloader.TrafficDataDownloaderLocal  
generalDAO=java:global/Verkeer2/GeneralDAO/GeneralDAO!iii.vop2016.verkeer2.  
   .ejb.dao.GeneralDAOLocal  
trafficDataDAO=java:global/Verkeer2/TrafficDataDAO/TrafficDataDAO!iii.vop2016.  
    verkeer2.ejb.dao.TrafficDataDAOLocal  
Timer=java:global/Verkeer2/TimerScheduler/TimerScheduler!iii.vop2016.verkeer2.  
   .ejb.timer.TimerSchedulerLocal  
GeoJsonProvider=java:global/Verkeer2/GeoJsonProvider/GeoJsonProvider!iii.  
    vop2016.verkeer2.ejb.geojson.GeoJsonLocal  
DataProvider=java:global/Verkeer2/DataProvider/DataProvider!iii.vop2016.  
    verkeer2.ejb.dataprovider.DataProviderLocal  
ThresholdManager=java:global/Verkeer2/Thresholds/ThresholdManager!iii.  
    vop2016.verkeer2.ejb.threshold.ThresholdManagerLocal  
Logger=java:global/Verkeer2/Logger/Logger!iii.vop2016.verkeer2.ejb.logger.  
    LoggerLocal  
LoginDAO=java:global/Verkeer2/LoginDAO/LoginDAO!iii.vop2016.verkeer2.ejb.  
    dao.LoginDAOLocal  
APIKeyDAO=java:global/Verkeer2/APIKeyDAO/APIKeyDAO!iii.vop2016.verkeer2.ejb.  
    datasources.SourceAdapterLocal  
Properties=java:global/Verkeer2/Properties/Properties
```

---

### CoyoteMapping

Het CoyoteMapping.properties-bestand bevat de mapping van de routenamen op Coyote met de namen van de routes op deze applicatie. De namen van de routes in deze applicatie zijn sleutels.

*Voorbeeld*

---

```
Rooigemlaan_(R40)_northbound=Rooigemlaan_(R40)_Northbound
```

---

## DataProvider

Het DataProvider.properties-bestand bevat informatie over de standaardintervallen en precisie die gebruikt worden wanneer er geen specifieke parameters zijn opgegeven bij de methodes in de DataProvider-klasse.

### Voorbeeld

---

```
OptimalDurationTimeFrame=1209600
OptimalDurationStartHour=06-00
OptimalDurationEndHour=07-00
AvgDurationTimeFrame=15379200
DataByDayTimeFrame=518400
TrendTimeFrame=1200
RecentDataTimeFrame=3660
DataTimeFrame=1209600
DataDefaultPrecision=100
```

---

## GeoJsonProvider

Het GeoJsonProvider.properties-bestand bevat een aantal zaken die gebruikt worden in de GeoJsonProvider-klasse om data op te halen.

### Voorbeeld

---

```
url=https://maps.googleapis.com/maps/api/directions/json
properties=mode,key
mode=driving
key=AIzaSyCYNBX_98P8_3P1hN_WBkuFrkxvVB9Ziks
```

---

## Logger

Het Logger.properties-bestand bevat informatie over de locatie en naam van het logbestand.

### Voorbeeld

---

```
filelocation=C:/
filename=log.xml
isactive=false
```

---

## Properties

Het Properties.properties-bestand bevat de defaultProviders die gebruikt worden in onder andere de DataProvider. Dit bestand wordt automatisch aangepast, wanneer een SourceAdapter wordt toegevoegd wordt de naam ervan automatisch toegevoegd.

*Voorbeeld*

---

```
defaultProviders=Here,Waze
```

---

## SourceAdapters

Het SourceAdapters.properties-bestand bevat verwijzingen naar de SourceAdapter-beans die gebruikt worden om RouteData op te halen. Elke adapter krijgt een uniek volgnummer mee in het bestand.

*Voorbeeld*

---

```
2=java:global/Verkeer2/GoogleMapsSourceAdapter/GoogleMapsSourceAdapter!iii.vop2016.verkeer2.ejb.datasources.SourceAdapterLocal
```

---

## SourceAdaptersKeys

Het SourceAdaptersKeys.properties-bestand bevat keys en andere informatie die gebruikt worden in de SourceAdapters om van de bronnen data op te halen.

*Voorbeeld*

---

```
GoogleMaps=xxx
```

---

## ThresholdHandlers

Het ThresholdHandlers.properties-bestand bevat verwijzingen naar de beans die gebruikt worden om bij bepaalde Thresholds meldingen te genereren.

*Voorbeeld*

---

```
twitter=java:global/Verkeer2/TwitterHandler/TwitterHandler!iii.vop2016.verkeer2.ejb.threshold.ThresholdHandlerLocal
```

---

## ThresholdManager

Het ThresholdManager.properties-bestand bevat de standaardwaarden die gebruikt worden om de Thresholds per Route te bepalen.

*Voorbeeld*

---

```
defaultThresholdLevels=120,240,480,1200
```

---



## TimerScheduler

Het TimerScheduler.properties-bestand bepaalt op welke tijdstippen het interval van de Timer gewijzigd moet worden. Het interval dat wordt meegegeven staat in minuten.

*Voorbeeld*

---

```
06-00=5
ntpserver=pool.ntp.org
bufferclear=05-30,10-30,13-30,15-30,19-30,22-30
```

---

## TrafficDataDownstreamAnalyser

Het TrafficDataDownstreamAnalyser.properties-bestand bepaalt of de filter voor het toevoegen van nieuwe data gebruikt wordt. Daarbij wordt ook nog de maximale afwijking op de afstand van de route in het data-object bepaald.

*Voorbeeld*

---

```
isDistanceFilterEnabled=true
distanceDif=1.1
```

---

## Twitter

Het Twitter.properties-bestand bevat informatie over het Twitter-account dat gebruikt wordt om meldingen te plaatsen.

*Voorbeeld*

---

```
baseUrl=https://api.twitter.com/1.1/statuses/update.json
requestMethode=POST
consumerkey=K7LONZJbBY57mYCPGvWyQkmfH
consumerSecret=3qKtLXwoz1KLUa1qPcqUBPg6HzvFijN7G1Ldw5ckMFm2x5PZWi
accessTokenSecret=OdhSNNUG0yDoJHouhFFNxp4usAm0UC6KCg5sdb40urAU3
signatureMethode=HMAC-SHA1
token=722691538124455937-rtWbuzGaZlVsDKeKGujgEBruXzdMP1q
version=1.0
imagePostUrl=https://upload.twitter.com/1.1/media/upload.json
imagePostRequestMethod=POST
media_type=image/png
imageRetrievalUrl=api.mapbox.com/v4/tobiasvdp.ac4aa6b2/geojson(GEOJSON)/..
twittermessageup=LEVEL verkeer gemeld op route ROUTE. De vertraging bedraagt
                    hier gemiddeld DELAYMINmDELAYSECs.
twittermessagedown=De verkeerssituatie op route ROUTE is aan het verbeteren.
                    De vertraging bedraagt hier gemiddeld nog DELAYMINmDELAYSECs.
```

---

## WazeMapping

Het WazeMapping.properties-bestand bevat de mapping van de routenamen op Waze met de namen van de routes op deze applicatie. De namen van de routes in deze applicatie zijn de sleutels.

### Voorbeeld

---

```
Rooigemlaan_(R40)_northbound=Rooigemlaan_(R40)_Northbound
```

---

## WebSettings

Het WebSettings.properties-bestand bepaalt de vorm van de URL's van de REST-aanvragen.

### Voorbeeld

---

```
apiKey=mike
urlREST=http://localhost:8080/RestApi/v2/
urlGeoJSONcurrent=http://localhost:8080/RestApi/v2/geojson/{id}/current
urlGeoJSONavg=http://localhost:8080/RestApi/v2/geojson/{id}/avg
urlRoutes=http://localhost:8080/RestApi/v2/routes/{id}
urlThresholds=http://localhost:8080/RestApi/v2/thresholds/{id}
urlTimerNewData=http://localhost:8080/RestApi/v2/timers/newdata
urlProviderComparer=http://localhost:8080/RestApi/v2/routes/{id}/
    providerDifference
urlPeriodComparer=http://localhost:8080/RestApi/v2/routes/{id}/
    periodDifference/{starts}/{ends}
urlAllProviders=http://localhost:8080/RestApi/v2/providers
urlAvgTraffic=http://localhost:8080/RestApi/v2/routes/{id}/avgTraffic
urlWeekdayComparer=http://localhost:8080/RestApi/v2/routes/{id}/days
urlRushhour=http://localhost:8080/RestApi/v2/routes/{id}/rushhours
urlServerLogs=http://localhost:8080/RestApi/v2/loggings?key={apikey}
urlUpdateRoute=http://localhost:8080/RestApi/v2/routes/{id}/update?
    key={apikey}
urlUpdateThresholds=http://localhost:8080/RestApi/v2/thresholds/update?
    key={apikey}
urlProperties=http://localhost:8080/RestApi/v2/settings/properties?
    key={apikey}
urlNewRoute=http://localhost:8080/RestApi/v2/routes/new?key={apikey}
urlRemoveRoute=http://localhost:8080/RestApi/v2/routes/{id}/remove?
    key={apikey}
urlUpdateProperties=http://localhost:8080/RestApi/v2/settings/properties?
    key={apikey}
urlApiKeys=http://verkeer-2.bp.tiwi.be/api/v2/settings/keys?
    key={apikey}
urlApiKeysInvalidate=http://verkeer-2.bp.tiwi.be/api/v2/settings/keys/
    invalidate?key={apikey}
urlApiKeysGenerate=http://verkeer-2.bp.tiwi.be/api/v2/settings/keys/generate?
    key={apikey}
urlGebruikers=http://verkeer-2.bp.tiwi.be/api/v2/settings/users?key={apikey}
urlGebruikersNieuw=http://verkeer-2.bp.tiwi.be/api/v2/settings/users/new?
    key={apikey}
AuthorizationBasic=cm9vdDpXYzdtaXVAcEE2
```

---

### 2.3.8 DataProvider - Methodes

DataProvider is het brein van de applicatie. In deze module worden alle aanvragen doorgestuurd naar de DAO's. Elke API-aanvraag steunt op DataProvider. Indien er bij een methode een aantal parameters niet meegegeven zijn zullen hiervoor standaardwaarden gebruikt worden, die terug te vinden zijn in het DataProvider.properties-bestand.

Hieronder worden in een paar hoofdstukken de methodes in tabelvorm besproken. In de eerste kolom staan de methodes en in de tweede kolom een korte uitleg over wat ze berekenen.

#### Elementaire methodes

Er zijn een aantal elementaire methodes waarvoor er weinig rekenkracht nodig is, deze hebben als *return*-waardes één getal.

Elementaire methodes	
getCurrentDuration/Velocity	gemiddelde reistijd/snelheid van data uit de laatste vijf minuten voor een bepaalde route en een aantal gespecificeerde providers
getOptimalDuration/Velocity	gemiddelde reistijd/snelheid in het optimale interval (gespecificeerd in het DataProvider.properties-bestand) voor een bepaalde route en een aantal gespecificeerde providers
getAvgDuration/Velocity	gemiddelde reistijd/snelheid in een bepaalde periode voor een route en een aantal gespecificeerde providers
get(Current/Avg)DelayLevel	vertragingen van een bepaalde route berekend over een bepaalde periode en voor een aantal gespecificeerde providers
getDistance	gemiddelde afstand van de meest recente RouteData voor een route en een aantal gespecificeerde providers
getTrend	een waarde die aangeeft of de vertraging van een route kleiner ( $x < 0$ ) of groter wordt ( $x > 0$ ); anders geeft dit de waarde 0 terug

## Uitgebreide methodes

De *return*-waardes voor de hieronder vermelde methodes zijn terug te vinden in de IDataProvider-interface in Verkeerlib.

Uitgebreide methodes	
getRecentData	gemiddelde reistijd in het laatste uur (standaardinstelling, wijzigbaar in DataProvider.properties) per <i>timestamp</i> voor één route en een aantal gespecificeerde providers
getData(Velocity)ByDay	gemiddelde reistijd per kwartier van 6u00 tot 2u00, bij 6u00 staat het gemiddelde van de data tussen 6u00 en 6u15 enzovoort
getData(Velocity)ByCombinedDay	neemt het gemiddelde uit DataByDay van de verschillende dagen in de periode die meegegeven wordt
getDataByDayHours	lijst met alle kwartieren tussen 6u en 2u 's nachts in deze formattering: 06-00 (=6u00)
getData(Velocity)	in een bepaalde periode afhankelijk van de precisie (aantal datapunten dat gevraagd wordt) verschillende timestamps met daarbij de gemiddelde reistijd, indien vanwege de precisie het interval tussen twee datapunten één uur bedraagt: bij 6u00 staat het gemiddelde van de data tussen 6u00 en 7u00 enzovoort
getMeanDurationFromRouteData	gewogen gemiddelde van de reistijd vanuit RouteData van verschillende providers
invalidateCurrentData	leegt de buffers (wordt aangeroepen door de timer na een tijd)

### 2.3.9 Databank

De database bestaat uit een aantal tabellen. RouteData is op termijn de grootste tabel. Hierin worden alle opgehaalde gegevens bewaard. In Routes staan alle trajecten waarvan data wordt opgehaald. Deze trajecten bestaan uit GeoLocations die het traject bepalen. Verder zijn er nog de Thresholds die bepalen wanneer een vertraging op een bepaalde route. Voor de communicatie- en presentatielaag is beveiliging nodig. Daarvoor dienen respectievelijk de APIKeys- en Userstabellen. In GeoLocationsMappings zit extra informatie over de GeoLocations.

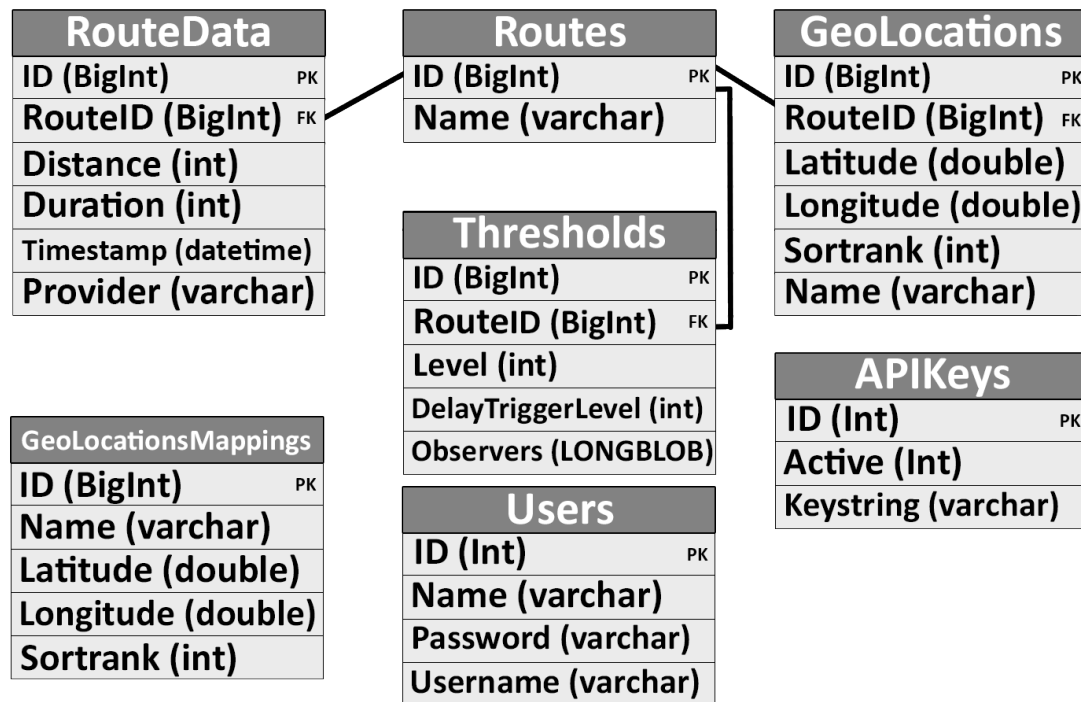


Diagram 14 Databank

### 2.3.10 Klassendiagram

In dit onderdeel worden drie klassendiagrammen beschreven, al kunnen het in dit geval ook interfacediagrammen genoemd worden. Om te beginnen is er een diagram voor de basiscomponenten, de meest elementaire klassen in het systeem. Deze klassen vormen ook de database. Hierna komt het diagram van het gegevensbeheer die de samenwerking van klassen die data ophalen en verwerken beschrijft. Als laatste, maar daarom niet minder belangrijk, bevindt zich de BeanFactory. Dit diagram bevat slechts één klasse die de klassen uit diagram 16 zal beheren.

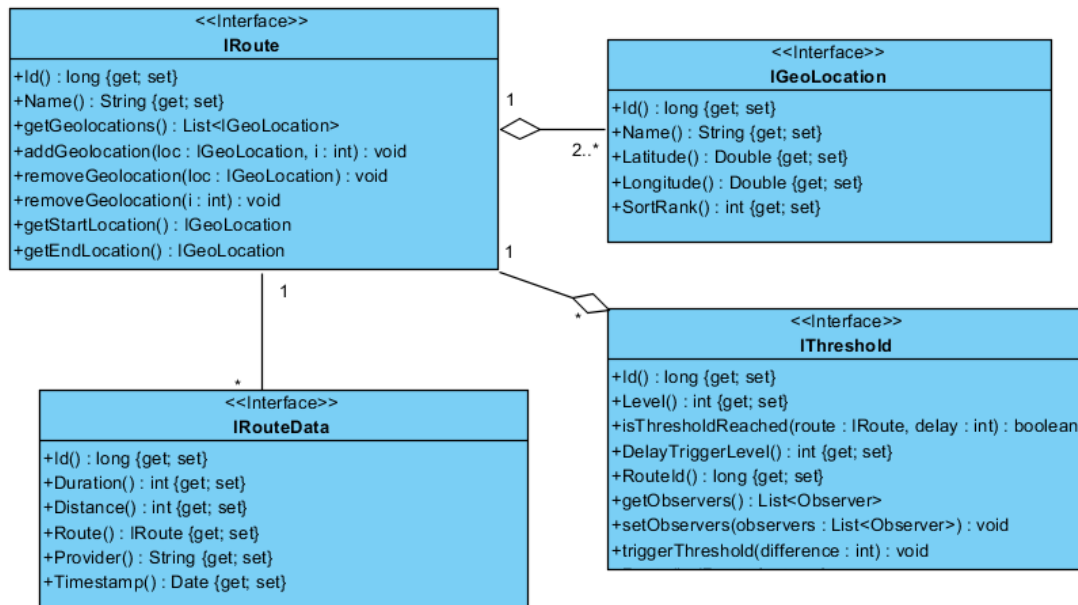


Diagram 15 Basiscomponenten

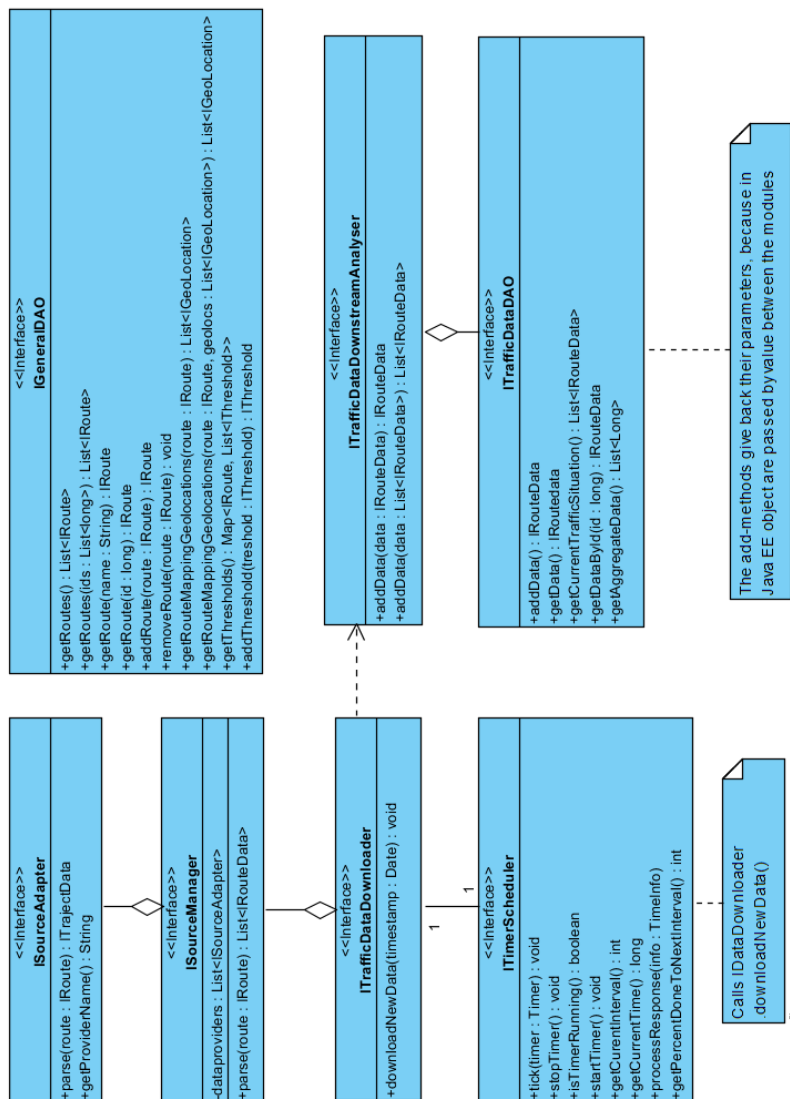


Diagram 16 Gegevensbeheer

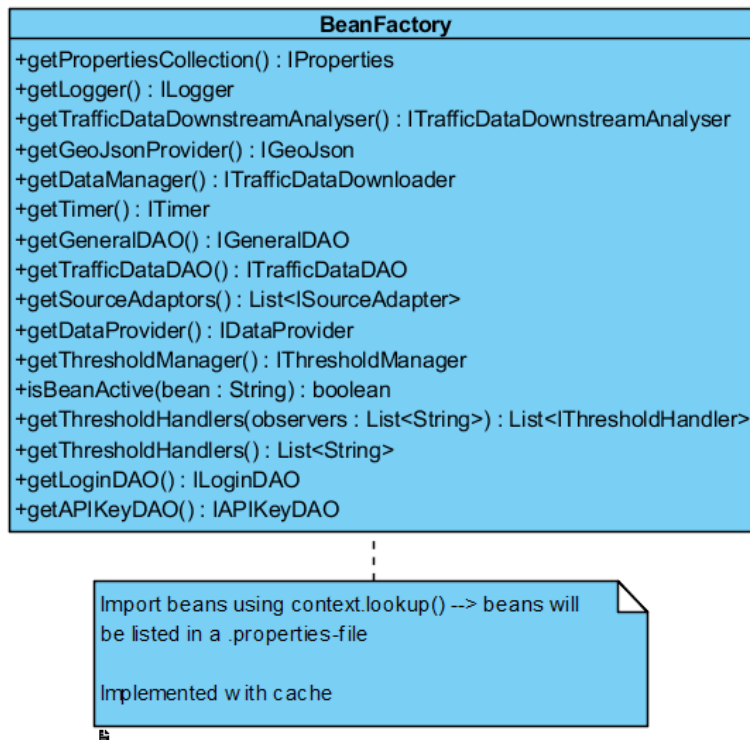


Diagram 17 BeanFactory



### 2.3.11 Gegevensstroomdiagram

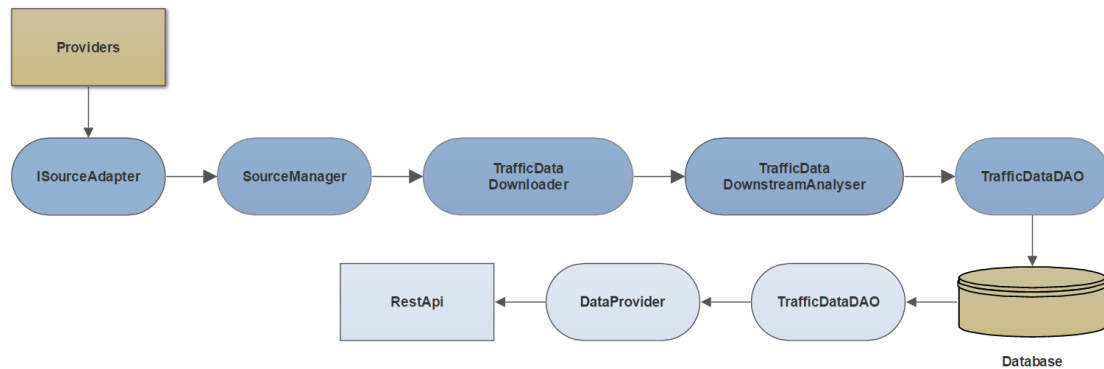


Diagram 18 Gegevensstroomdiagram

#### Downstream

De data van de verschillende databronnen wordt opgehaald met behulp van de SourceAdapters. Per databron bestaat er een adapterklasse die de ISourceAdapter-interface implementeert. De verschillende adapters bevinden zich in de SourceManager-klasse. Vanuit de TrafficDataDownloader wordt data per route aangevraagd. De SourceManager zal deze aanvragen doorsturen aan elke adapter en de ontvangen data per adapter teruggeven aan de TrafficDataDownloader. Vervolgens stromen de gegevens door de TrafficDataDownstreamAnalyser die eventueel meldingen genereert en controleert of de data geldig is. Om af te sluiten wordt de data doorgegeven aan de TrafficDataDAO die ervoor zorgt dat de data in de database terechtkomt.

#### Upstream

Via API-aanvragen kan data uit de database gehaald worden. Vooreerst gebeurt dit via de TrafficDataDAO die contact heeft met de database. Via de DataProvider wordt een bepaalde aanvraag gedaan aan de TrafficDataDAO. De RESTAPI zorgt er dan voor dat de *return*-waarden van de DataProvider in JSON-formaat worden omgezet.

### **2.3.12 Line Of Business**

De volledige applicatie streeft naar de richtlijnen van een Line Of Business applicatie.

#### **Flexibel en Uitbreidbaar**

Door gebruik te maken van Java EE, waarin de gehele applicatie in verschillende modules wordt opgedeeld, kunnen nieuwe modules eenvoudig afzonderlijk worden gecreëerd en worden toegevoegd.

#### **Onderhoudbaarheid**

Dit analysedocument bevat alle nodige informatie over de klassen en hun onderlinge relaties. Op het einde van de ontwikkelingsperiode zal een documentatie worden voorzien met alle nodige informatie voor andere ontwikkelaars die de applicatie zouden willen wijzigen of uitbreiden.

#### **Testbaarheid**

Een aantal componenten werden getest aan de hand van unittests, in een latere fase werden deze tests echter onmogelijk. Door de modulariteit in combinatie met het gebruik van properties-bestanden moet de server altijd aanstaan voor er tests uitgevoerd kunnen worden.

Door deze problemen werden de belangrijkste zaken handmatig gecontroleerd. Meer uitgebreide info over de tests is terug te vinden in het testplan.

#### **Late Binding**

Java EE biedt de mogelijkheid om een applicatie op te delen in verschillende modules die afzonderlijk van elkaar kunnen worden gecompileerd. Er werden twee DAO's voorzien zodat de algemene data over routes en de verkeersinformatie over de routes kunnen worden opgeslagen in twee verschillende databases. Zo zal na de ontwikkelingsperiode de verkeersinformatie waarschijnlijk worden opgeslagen in een NoSQL-database, omdat de hoeveelheid data enorm groot zal worden.

#### **Parallele ontwikkeling**

Door opdeling in modules, die Java EE aanbiedt, kunnen programmeurs afzonderlijk van elkaar code implementeren.

#### **Losse koppeling van objecten**

Modules kunnen eenvoudig worden ontkoppeld en worden vervangen door een andere. De BeanFactory, die de module-objecten aanbiedt, wordt eenvoudig geconfigureerd in een properties-bestand. Zo kan eenvoudig een nieuwe DAO worden toegevoegd door de link in het configuratiebestand te wijzigen naar een andere DAO. Zo kan bijvoorbeeld de manier van opslaan van data eenvoudig worden gewijzigd van een SQL-database naar een NoSQL-database.

#### **Crosscutting concerns (Logging)**

De manier van logging kan eenvoudig worden gewijzigd door het hierboven beschreven principe van losse koppeling. Voor logging werd ook een module (Logger) voorzien waarnaar de verwijzing kan worden veranderd door de link aan te passen in het Beans.properties-bestand.

## 2.4 DATA ONTWERP

De RESTAPI geeft info terug uit de database, hier heb je de keuze uit verschillende opties die in de volgende delen besproken worden. Wanneer je hier aanvragen naar doet krijg je gegevens uit de database terug. In de komende secties wordt een korte toelichting gegeven per optie, gevolg door een voorbeeld van aanvraag en antwoord in JSON-formaat.

### 2.4.1 Routes

#### Algemeen

De routes zijn de trajecten waar realtime data over opgeroepen wordt. Via deze aanvraag wordt alle statische informatie zoals naam, id en tussenpunten meegegeven. Maar ook dynamische informatie, specifiek voor deze route zoals optimale, actuele en gemiddelde snelheid alsook trend en vertragsingslevel worden in het antwoord opgenomen.

#### Vraag

*Patroon*

**GET** <http://verkeer2.bp.tiwi.be/RestApi/v2/routes/id>

Parameter(s)	Beschrijving
id	lijst van route-id's gescheiden door een komma, ook het woord <i>all</i> is toegestaan

Optionele parameter(s)	Beschrijving
providers	Lijst van providers (gescheiden door komma's) die aangeeft welke providers in de berekeningen gebruikt mogen worden. Defaultwaarde: all.
avgstart	Start van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
avgend	Einde van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
optimalstart	Start van het interval waarover je de optimale snelheid en reistijd wil berekenen.
optimalend	Einde van het interval waarover je de optimale snelheid en reistijd wil berekenen.

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/2?providers=GoogleMaps,Here&avgstart=1456761535931&avgend=1457829774992&optimalstart=1456761535931&optimalend=1457829774992>

#### Antwoord

*Voorbeeld*

```
[
{
  "id": 2,
  "name": "R4 Zelzate",
  "geolocations": [
    {
      "name": "Zelzate",
      "id": 20,
      "sortRank": 1,
      "latitude": 51.192226,
```

```

    "longitude": 3.776342
  },
  {
    "name": "Gent",
    "id": 21,
    "sortRank": 2,
    "latitude": 51.086447,
    "longitude": 3.672188
  }
],
"currentDuration": 753,
"currentVelocity": 95,
"optimalDuration": 667,
"optimalVelocity": 120,
"avgDuration": 745,
"avgVelocity": 99,
"trend": 0,
"recentData": {
  "duration": {
    "name": "recentData 2",
    "description": "This data are the durations over the last hour for route 2",
    "data": {
      "1456761535931": 703,
      "1456761537931": 709,
      "1456761539931": 709,
      "1456761541931": 711,
      "1456761543931": 711
    }
  }
},
"currentDelayLevel": 2,
"avgDelayLevel": 2,
"distance": 14677
}
]

```

---

## Ruwe data

Deze aanvraag zal de data van de verschillende databronnen weergeven.

### Vraag

*Patroon*

**GET** <http://verkeer2.bp.tiwi.be/RestApi/v2/routes/id/raw>

Parameter(s)	Beschrijving
id	lijst van route-id's gescheiden door een komma, ook het woord <i>all</i> is toegestaan

Optionele parameter(s)	Beschrijving
providers	Lijst van providers (gescheiden door komma's) die aangeeft welke providers in de berekeningen gebruikt mogen worden. Defaultwaarde: all.
avgstart	Start van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
avgend	Einde van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
optimalstart	Start van het interval waarover je de optimale snelheid en reistijd wil berekenen.
optimalend	Einde van het interval waarover je de optimale snelheid en reistijd wil berekenen.
start	Start van het interval waarvan je data wil verkrijgen.
end	Einde van het interval waarvan je data wil verkrijgen.
page	paginering in de records, pagina 0 geeft de eerste 1000 records enzovoort

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/1/raw?providers=GoogleMaps,Here&avgstart=1456761535931&avgend=1457829774992&optimalstart=1456761535931&optimalend=1457829774992&start=1446761535931&end=1487829774992&key=YourActivatedAPIKey&page=0>

### Antwoord

*Voorbeeld*

```
[
{
  "id": 2,
  "name": "R4 Zelzate",
  "geolocations": [
    {
      "name": "Zelzate",
      "id": 20,
      "sortRank": 1,
      "latitude": 51.192226,
      "longitude": 3.776342
    },
    {
      "name": "Gent",
      "id": 21,
      "sortRank": 2,
      "latitude": 51.086447,
```

```

    "longitude": 3.672188
  }
],
"currentDuration": 753,
"currentVelocity": 7,
"optimalDuration": 667,
"optimalVelocity": 9,
"avgDuration": 745,
"avgVelocity": 8,
"trend": 0,
"recentData": {
  "duration": {
    "name": "recentData 2",
    "description": "This data are the durations over the last hour for route 2",
    "data": {
      "1456761535931": 703,
      "1456761537931": 709,
      "1456761539931": 709,
      "1456761541931": 711,
      "1456761543931": 711
    }
  }
},
"currentDelayLevel": 2,
"avgDelayLevel": 2,
"distance": 14677,
"rawdata": [
  {
    "duration": 148,
    "routeId": 1,
    "distance": 1345,
    "provider": "GoogleMaps",
    "id": 41,
    "timestamp": 1461691169000
  },
  {
    "duration": 190,
    "routeId": 1,
    "distance": 1351,
    "provider": "Here",
    "id": 60,
    "timestamp": 1461691169000
  },
  {
    "duration": 144,
    "routeId": 1,
    "distance": 1345,
    "provider": "GoogleMaps",
    "id": 109,
    "timestamp": 1461691469000
  },
  {
    "duration": 189,
    "routeId": 1,
    "distance": 1351,

```

```
"provider": "Here",  
"id": 123,  
"timestamp": 1461691469000  
}  
]  
}  
]
```

---

## 2.4.2 Analyses

In deze sectie komende de verschillende analyses aan bod, elk van deze analyses legt de focus op een bepaald aspect van de data.

### Data per dag

In deze analyse krijg je tussen start en eindpunt per dag (maandag tot en met zondag) de gemiddelde reistijd en snelheid, aangevuld met extra informatie over de huidige situatie op de route.

---

#### Vraag

*Patroon*

**GET** <http://verkeer2.bp.tiwi.be/RestApi/v2/routes/id/days>

Parameter(s)	Beschrijving
id	lijst van route-id's gescheiden door een komma, ook het woord <i>all</i> is toegestaan

Optionele parameter(s)	Beschrijving
providers	Lijst van providers (gescheiden door komma's) die aangeeft welke providers in de berekeningen gebruikt mogen worden. Defaultwaarde: all.
avgstart	Start van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
avgend	Einde van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
optimalstart	Start van het interval waarover je de optimale snelheid en reistijd wil berekenen.
optimalend	Einde van het interval waarover je de optimale snelheid en reistijd wil berekenen.
start	Start van het interval waarvan je data wil verkrijgen.
end	Einde van het interval waarvan je data wil verkrijgen.

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/1/days?providers=GoogleMaps,Here&avgstart=1456761535931&avgend=1457829774992&optimalstart=1456761535931&optimalend=1457829774992&start=1456761535931&end=1457829774992>

#### Antwoord

*Voorbeeld*

```
[
{
  distance: 1348,
  data: {
    duration: {
      data: {
        WEDNESDAY: [
          151,
          146,
          150,
          149,
          157,
          156,
```



```

157,
167,
174,
179,
168,
172,
171,
164,
158,
165,
164,
-1,
164,
187,
170,
171,
162,
188,
192,
205,
207,
209,
174,
175,
.
.
.
],
MONDAY: [...],
THURSDAY: [...],
SUNDAY: [...],
TUESDAY: [...],
FRIDAY: [...],
SATURDAY: [...],
x-ax: [
"06:00",
"06:15",
"06:30",
"06:45",
"07:00",
"07:15",
"07:30",
"07:45",
"08:00",
"08:15",
"08:30",
"08:45",
"09:00",
"09:15",
"09:30",
"09:45",
"10:00",
"10:15",
"10:30",
"10:45",

```

```

        "11:00",
        "11:15",
        "11:30",
        "11:45",
        "12:00",
        "12:15",
        "12:30",
        "12:45",
        "13:00",
        "13:15",
        .
        .
        .
    ]
},
name: "dayDataDurations 1",
description: "This data are the durations for every day of the week for route 1"
},
velocity: {}
},
trend: 0,
currentDelayLevel: 0,
optimalDuration: 145,
avgDuration: 163,
currentVelocity: 6,
avgDelayLevel: 0,
recentData: {..},
avgVelocity: 8,
optimalVelocity: 9,
name: "Rooigemlaan (R40) northbound",
currentDuration: 185,
id: 1,
geolocations: [..]
}
]

```

---

## Spitsuren tijdens werkdagen

Het gemiddelde van de gegevens op een werkdag (maandag tot en met vrijdag) in een bepaalde periode.

---

### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/id/rushhours>

Parameter(s)	Beschrijving
id	lijst van route-id's gescheiden door een komma, ook het woord <i>all</i> is toegestaan

Optionele parameter(s)	Beschrijving
providers	Lijst van providers (gescheiden door komma's) die aangeeft welke providers in de berekeningen gebruikt mogen worden. Defaultwaarde: all.
avgstart	Start van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
avgend	Einde van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
optimalstart	Start van het interval waarover je de optimale snelheid en reistijd wil berekenen.
optimalend	Einde van het interval waarover je de optimale snelheid en reistijd wil berekenen.
start	Start van het interval waarover je data wil verkrijgen.
end	Einde van het interval waarover je data wil verkrijgen.

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/1/rushhours?providers=GoogleMaps,Here&avgstart=1456761535931&avgend=1457829774992&optimalstart=1456761535931&optimalend=1457829774992&start=1456761535931&end=1457829774992>

### Antwoord

*Voorbeeld*

```
[
{
  distance: 1347,
  data: {
    duration: {
      data: {
        workday: [
          122,
          125,
          131,
          131,
          137,
          138,
          148,
          155,
          .
          .
        ]
      }
    }
  }
}
```

```

    .
  ],
  x-ax: [
    "06:00",
    "06:15",
    "06:30",
    "06:45",
    "07:00",
    "07:15",
    "07:30",
    "07:45",
    .
    .
    .
  ]
},
name: "rushhourDurations 1",
description: "This data are the durations for a workday for route 1"
},
velocity: {...}
},
trend: 0,
currentDelayLevel: 0,
optimalDuration: 127,
avgDuration: 142,
currentVelocity: 7,
avgDelayLevel: 0,
recentData: {...},
avgVelocity: 11,
optimalVelocity: 13,
name: "Rooigemlaan (R40) northbound",
currentDuration: 179,
id: 1,
geolocations: [...]
}
]

```

---

## Verschil tussen periodes

In deze analyse worden meerdere periodes met elkaar vergeleken.

### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/id/periodDifference/startTimes/endTimes>

Parameter(s)	Beschrijving
id	lijst van route-id's gescheiden door een komma, ook het woord <i>all</i> is toegestaan
startTimes	Startmomenten van de periodes, gescheiden door komma's.
endTimes	Eindmomenten van de periodes, gescheiden door komma's.

Optionele parameter(s)	Beschrijving
providers	Lijst van providers (gescheiden door komma's) die aangeeft welke providers in de berekeningen gebruikt mogen worden. Defaultwaarde: all.
avgstart	Start van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
avgend	Einde van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
optimalstart	Start van het interval waarover je de optimale snelheid en reistijd wil berekenen.
optimalend	Einde van het interval waarover je de optimale snelheid en reistijd wil berekenen.
precision	Hoeveel records wil je (maximaal) per periode terugkrijgen.

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/2/periodDifference/1456761535931,1457761535931/1456861535931,1457861535931?providers=GoogleMaps,Here&avgstart=1456761535931&avgend=1457829774992&optimalstart=1456761535931&optimalend=1457829774992&precision=100>

### Antwoord

*Voorbeeld*

```
[
{
  "id": 2,
  "name": "R4 Zelzate",
  "geolocations": [..],
  "currentDuration": 753,
  "currentVelocity": 9,
  "optimalDuration": 667,
  "optimalVelocity": 8,
  "avgDuration": 745,
  "avgVelocity": 8,
  "trend": 0,
  "recentData": {..},
  "currentDelayLevel": 2,
  "avgDelayLevel": 2,
  "distance": 14677,
  "data": [
```

```

{
  "period": "period1",
  "start": 1456761535931,
  "end": 1456861535931,
  "data": {
    "duration": {
      "name": "period1Durations 2",
      "description": "This data are the durations for period 1 for route 2",
      "data": {
        "1456761535931": 703,
        "1456761537931": 709,
        "1456761539931": 709,
        "1456761541931": 711,
        "1456761543931": 711
      }
    },
    "velocity": {
      "name": "period1Velocities 2",
      "description": "This data are the velocities for period 1 for route 2",
      "data": {
        "1456761535931": 7,
        "1456761537931": 9,
        "1456761539931": 9,
        "1456761541931": 11,
        "1456761543931": 7
      }
    }
  }
},
{
  "period": "period2",
  "start": 1457761535931,
  "end": 1457861535931,
  "data": {
    "duration": {
      "name": "period2Durations 2",
      "description": "This data are the durations for period 2 for route 2",
      "data": {
        "1456761535931": 703,
        "1456761537931": 709,
        "1456761539931": 709,
        "1456761541931": 711,
        "1456761543931": 711
      }
    },
    "velocity": {
      "name": "period2Velocities 2",
      "description": "This data are the velocities for period 2 for route 2",
      "data": {
        "1456761535931": 7,
        "1456761537931": 9,
        "1456761539931": 9,
        "1456761541931": 7,
        "1456761543931": 7
      }
    }
  }
}

```

```
}  
}  
}  
]  
}  
]
```

---

## Verschil tussen databronnen

In deze analyse worden meerdere databronnen met elkaar vergeleken.

### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/id/providerDifference/>

Parameter(s)	Beschrijving
id	lijst van route-id's gescheiden door een komma, ook het woord <i>all</i> is toegestaan

Optionele parameter(s)	Beschrijving
providers	Lijst van providers (gescheiden door komma's) die aangeeft welke providers in de berekeningen gebruikt mogen worden. Defaultwaarde: all.
avgstart	Start van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
avgend	Einde van het interval waarover je de gemiddelde snelheid en reistijd wil berekenen.
optimalstart	Start van het interval waarover je de optimale snelheid en reistijd wil berekenen.
optimalend	Einde van het interval waarover je de optimale snelheid en reistijd wil berekenen.
start	Start van het interval waarover je data wil verkrijgen.
end	Einde van het interval waarover je data wil verkrijgen.
precision	Hoeveel records wil je (maximaal) per provider terugkrijgen.

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/routes/2/providerDifference?providers=GoogleMaps,Here&avgstart=1456761535931&avgend=1457829774992&optimalstart=1456761535931&optimalend=1457829774992&start=1456761535931&end=1457829774992&precision=100>

### Antwoord

*Voorbeeld*

```
[
{
  "id": 2,
  "name": "R4 Zelzate",
  "geolocations": [..],
  "currentDuration": 753,
  "currentVelocity": 8,
  "optimalDuration": 667,
  "optimalVelocity": 9,
  "avgDuration": 745,
  "avgVelocity": 8,
  "trend": 0,
  "recentData": {..},
  "currentDelayLevel": 2,
  "avgDelayLevel": 2,
  "distance": 14677,
  "data": [
```



```

{
  "provider": "GoogleMaps",
  "data": {
    "duration": {
      "name": "providerDurations GoogleMaps 2",
      "description": "This data are the durations provided by GoogleMaps
                      for route 2",
      "data": {
        "1456761535931": 703,
        "1456761537931": 709,
        "1456761539931": 709,
        "1456761541931": 711,
        "1456761543931": 711
      }
    },
    "velocity": {...}
  },
  {
    "provider": "Here",
    "data": {
      "duration": {...},
      "velocity": {...}
    }
  }
]
}
]

```

---

### 2.4.3 Providers

Het is mogelijk om via de RESTAPI alle databronnen op te vragen, dit kan handig zijn om bijvoorbeeld in de analyses te gebruiken als optionele parameter.

---

#### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/providers>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/providers>

#### Antwoord

*Voorbeeld*

```
[  
  "Here",  
  "GoogleMaps",  
  "Coyote",  
  "TomTom",  
  "Waze"  
]
```

---

### 2.4.4 API Keys

Hieronder zullen de verschillende aanvragen met betrekking tot de API Keys kort besproken worden.

#### Alle API Keys

In onderstaande API aanvraag zullen alle bestaande API keys teruggegeven worden.

---

##### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/keys>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/keys?key=YourActivatedAPIKey>

##### Antwoord

*Voorbeeld*

```
[
  {
    "active": 1,
    "id": 2,
    "key": "aKeyWithCharsAndNumbers"
  },
  {
    "active": 0,
    "id": 3,
    "key": "aSecondKeyWithCharsAndNumbers"
  }
]
```

---

#### Genereer een API Key

Aan de hand van deze aanvraag zal een nieuwe API key gegenereerd worden.

---

##### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/generate>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/generate?key=YourActivatedAPIKey>

##### Antwoord

*Voorbeeld*

```
[
  {
    "active": 1,
    "id": 2,
    "key": "aKeyWithCharsAndNumbers"
  }
]
```

## Invalideer een API Key

Aan de hand van deze aanvraag zal een bestaande API key ongeldig gemaakt worden.

---

### Vraag

*Patroon*

**POST** http://verkeer-2.bp.tiwi.be/api/v2/settings/**invalidate**

*Voorbeeld*

**POST** http://verkeer-2.bp.tiwi.be/api/v2/settings/**invalidate?**  
**key=YourActivatedAPIKey**

### Aanvraag

*Voorbeeld*

```
{  
  "active": 1,  
  "id": 2,  
  "key": "aKeyWithCharsAndNumbers"  
}
```

### Antwoord

*Voorbeeld*

```
{  
  "status": "done"  
}
```

---

### 2.4.5 Gebruikers

Hieronder zullen de verschillende aanvragen met betrekking tot de gebruikers (op de webapplicatie) kort besproken worden.

#### Lijst van alle gebruikers

Deze aanvraag zal een lijst van alle gebruikers teruggeven.

---

##### Vraag

*Patroon*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/settings/users`

*Voorbeeld*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/settings/users?key=YourActivatedAPIKey`

##### Antwoord

*Voorbeeld*

```
[
  {
    "name": "Danil Termont",
    "id": 1,
    "username": "danieltermont"
  },
  {
    "name": "Barack Obama",
    "id": 2,
    "username": "presidentobama"
  }
]
```

---

#### Een specifieke gebruiker

Deze aanvraag zal een lijst van alle gebruikers teruggeven.

---

##### Vraag

*Patroon*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/settings/users/{id}`

*Voorbeeld*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/settings/users/1?key=YourActivatedAPIKey`

##### Antwoord

*Voorbeeld*

```
{
  "name": "Danil Termont",
  "id": 1,
  "username": "danieltermont"
}
```

---

## Nieuwe gebruiker

Op deze manier wordt een nieuwe gebruiker aangemaakt.

---

### Vraag

*Patroon*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/settings/users/new>

*Voorbeeld*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/settings/users/new?key=YourActivatedAPIKey>

### Aanvraag

*Voorbeeld*

```
{
  "name": "Hein Vanhaezebrouck",
  "id": 0,
  "username": "trainerhein",
  "password": "KAAGent"
}
```

### Antwoord

*Voorbeeld*

```
{
  "status": "done"
}
```

---

## Update gebruiker

Op deze manier wordt een gebruiker aangepast.

---

### Vraag

*Patroon*

**POST** `http://verkeer-2.bp.tiwi.be/api/v2/settings/users/update/{id}`

*Voorbeeld*

**POST** `http://verkeer-2.bp.tiwi.be/api/v2/settings/users/update/1?  
key=YourActivatedAPIKey`

### Aanvraag

*Voorbeeld*

```
{  
  "name": "Danil Termont",  
  "id": 1,  
  "username": "burgemeesterdaniel",  
  "password": "Password or empty if you don't want to change it"  
}
```

### Antwoord

*Voorbeeld*

```
{  
  "status": "done"  
}
```

---

## 2.4.6 Thresholds

### Thresholds voor alle routes

In deze aanvraag worden alle thresholds van alle routes teruggegeven.

---

#### Vraag

*Patroon*

**GET** http://verkeer-2.bp.tiwi.be/api/v2/thresholds

*Voorbeeld*

**GET** http://verkeer-2.bp.tiwi.be/api/v2/thresholds

#### Antwoord

*Voorbeeld*

```
[
  {
    "thresholds": [
      {
        "routeId": 26,
        "level": 1,
        "delayTrigger": 120,
        "handlers": [],
        "id": 102
      },
      {
        "routeId": 26,
        "level": 2,
        "delayTrigger": 240,
        "handlers": [],
        "id": 104
      },
      {
        "routeId": 26,
        "level": 3,
        "delayTrigger": 480,
        "handlers": [],
        "id": 103
      },
      {
        "routeId": 26,
        "level": 4,
        "delayTrigger": 1200,
        "handlers": [],
        "id": 101
      }
    ],
    "name": "Antwerpsesteenweg (N70) eastbound",
    "id": 26,
    "geolocations": [...]
  },
  {...}..
]
```

---



## Thresholds voor specifieke routes

Deze aanvraag geeft de thresholds voor één of meerdere routes terug.

---

### Vraag

*Patroon*

**GET** http://verkeer-2.bp.tiwi.be/api/v2/thresholds/{id}

*Voorbeeld*

**GET** http://verkeer-2.bp.tiwi.be/api/v2/thresholds/26

### Antwoord

*Voorbeeld*

```
[
  {
    "thresholds": [
      {
        "routeId": 26,
        "level": 1,
        "delayTrigger": 120,
        "handlers": [],
        "id": 102
      },
      {
        "routeId": 26,
        "level": 2,
        "delayTrigger": 240,
        "handlers": [],
        "id": 104
      },
      {
        "routeId": 26,
        "level": 3,
        "delayTrigger": 480,
        "handlers": [],
        "id": 103
      },
      {
        "routeId": 26,
        "level": 4,
        "delayTrigger": 1200,
        "handlers": [],
        "id": 101
      }
    ],
    "name": "Antwerpsesteenweg (N70) eastbound",
    "id": 26,
    "geolocations": [...]
  },
  {...}..
]
```

---

## Thresholdhandlers

Deze aanvraag zal een lijst van alle mogelijke thresholdhandlers teruggeven.

---

### Vraag

*Patroon*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/threshold/handlers`

*Voorbeeld*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/threshold/handlers`

### Antwoord

*Voorbeeld*

```
[  
  "twitter"  
]
```

---

## 2.4.7 Timer

### Informatie over de timer

Deze aanvraag bevat informatie over de huidige situatie van de timer.

---

#### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/timers/newdata>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/timers/newdata>

#### Antwoord

*Voorbeeld*

```
{
  "percentDone": 80,
  "active": "true",
  "interval": 300,
  "time": 1469864561
}
```

---

### Start de timer

Via deze aanvraag kan de timer gestart worden.

---

#### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/timers/start>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/timers/start?key=YourActivatedAPIKey>

#### Antwoord

*Voorbeeld*

```
{
  "status": "ok"
}
```

---

## Stop de timer

Via deze aanvraag kan de timer gestopt worden.

---

### Vraag

*Patroon*

**GET** http://verkeer-2.bp.tiwi.be/api/v2/timers/**stop**

*Voorbeeld*

**GET** http://verkeer-2.bp.tiwi.be/api/v2/timers/**stop?key=YourActivatedAPIKey**

### Antwoord

*Voorbeeld*

```
{  
  "status": "ok"  
}
```

---

## 2.4.8 Loggings

Deze aanvraag zal de serverlogs teruggeven aan de hand van een aantal parameters.

---

### Vraag

*Patroon*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/loggings`

Optionele parameter(s)	Beschrijving
key	Een werkende API Key.
amount	Het aantal records dat je wil terugkrijgen.
offset	Geeft het aantal logs terug vanaf deze waarde, indien dit bijvoorbeeld 50 is worden de meest recente 50 logs niet in rekening gebracht.
filter	Maakt het mogelijk bepaalde logs te selecteren, opties: severe, warning, info, config, fine, finer and finest.
containing	Een string die de logs moeten bevatten.

*Voorbeeld*

**GET** `http://verkeer-2.bp.tiwi.be/api/v2/loggings?key=YourActivatedAPIKey&amount=1&offset=0&filter=finer&containing=data`

### Antwoord

*Voorbeeld*

```
[
  {
    "date": 1462634933704,
    "level": "FINER",
    "message": "Data scrubbing done"
  }
]
```

---

### 2.4.9 GeoJson

GeoJson bevat de verschillende geolocaties die nodig zijn om op een kaart (bijvoorbeeld op de webapplicatie) de routes te tekenen.

#### Huidige GeoJson

De waarden voor de live-weergave.

---

#### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/geojson/id/current>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/geojson/1/current>

#### Antwoord

*Voorbeeld*

```
{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        3.69516,
        51.05611
      ],
      [..],
      [..],
      .
      .
      .
    ]
  },
  "properties": {
    "id": 1,
    "currentDelayLevel": 0
  }
}
```

---

## Gemiddelde GeoJson

De waarden voor de weergave van de gemiddelde situatie.

---

### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/geojson/id/average>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/geojson/1/average>

### Antwoord

*Voorbeeld*

```
{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        3.69516,
        51.05611
      ],
      [..],
      [..],
      .
      .
      .
    ]
  },
  "properties": {
    "id": 1,
    "currentDelayLevel": 0
  }
}
```

---

## 2.4.10 Instellingen

### Initialiseer alle routes

Deze opdracht kan gebruikt worden om de 34 standaardroutes te initialiseren.

---

#### Vraag

*Patroon*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/routes/init>

*Voorbeeld*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/routes/init?key=YourActivatedAPIKey>

#### Antwoord

*Voorbeeld*

Status: 200

---

### Nieuwe route

Deze opdracht kan gebruikt worden om een nieuwe route aan te maken.

---

#### Vraag

*Patroon*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/routes/new>

*Voorbeeld*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/routes/new?key=YourActivatedAPIKey>

#### Aanvraag

*Voorbeeld*

```
{
  "name": "R4 Zelzate",
  "id": 0,
  "geolocations": [
    {
      "name": "Zelzate",
      "id": 0,
      "sortRank": 1,
      "latitude": 51.192226,
      "longitude": 3.776342
    },
    {
      "name": "Gent",
      "id": 0,
      "sortRank": 2,
      "latitude": 51.086447,
      "longitude": 3.672188
    }
  ]
}
```



### Antwoord

*Voorbeeld*

```
{
  "status": "ok",
  "message": "Route has been created"
}
```

---

### Verwijder route

Deze opdracht kan gebruikt worden om een route te verwijderen.

---

### Vraag

*Patroon*

**POST** [http://verkeer-2.bp.tiwi.be/api/v2/routes/\*\*id\*\*/remove](http://verkeer-2.bp.tiwi.be/api/v2/routes/id/remove)

*Voorbeeld*

**POST** [http://verkeer-2.bp.tiwi.be/api/v2/routes/\*\*1\*\*/remove?key=YourActivatedAPIKey](http://verkeer-2.bp.tiwi.be/api/v2/routes/1/remove?key=YourActivatedAPIKey)

### Antwoord

*Voorbeeld*

```
{
  "status": "ok",
  "message": "Route has been removed"
}
```

---

## Update route

Deze opdracht kan gebruikt worden om een nieuwe route te wijzigen.

---

### Vraag

*Patroon*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/routes/id/update>

*Voorbeeld*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/routes/1/update?key=YourActivatedAPIKey>

### Aanvraag

*Voorbeeld*

```
{
  "name": "R4 Zelzate",
  "id": 1,
  "geolocations": [
    {
      "name": "Zelzate",
      "id": 20,
      "sortRank": 1,
      "latitude": 51.192226,
      "longitude": 3.776342
    },
    {
      "name": "Gent",
      "id": 21,
      "sortRank": 2,
      "latitude": 51.086447,
      "longitude": 3.672188
    }
  ]
}
```

### Antwoord

*Voorbeeld*

```
{
  "status": "ok",
  "message": "Route has been updated"
}
```

---

## Update thresholds

Deze opdracht kan gebruikt worden om thresholds te wijzigen.

---

### Vraag

*Patroon*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/thresholds/update>

*Voorbeeld*

**POST** <http://verkeer-2.bp.tiwi.be/api/v2/thresholds/update?key=YourActivatedAPIKey>

### Aanvraag

*Voorbeeld*

```
[
{
  "thresholds": [
    {
      "routeId": 1,
      "level": 1,
      "delayTrigger": 120,
      "handlers": [],
      "id": 1
    },
    {
      "routeId": 1,
      "level": 2,
      "delayTrigger": 240,
      "handlers": [],
      "id": 4
    },
    {
      "routeId": 1,
      "level": 3,
      "delayTrigger": 480,
      "handlers": [],
      "id": 2
    },
    {
      "routeId": 1,
      "level": 4,
      "delayTrigger": 1200,
      "handlers": [],
      "id": 3
    }
  ],
  "name": "Rooigemlaan (R40) northbound",
  "id": 1,
  "geolocations": [
    {
      "sortRank": 2,
      "latitude": 51.066296,
      "name": "Palinghuizen",
      "id": 1,

```

```
    "longitude": 3.699685
  },
  {
    "sortRank": 1,
    "latitude": 51.056104,
    "name": "Drongensesteenweg",
    "id": 2,
    "longitude": 3.695152
  }
]
}
```

### Antwoord

*Voorbeeld*

```
{
  "status": "success"
}
```

---

### Buffers leegmaken

Met deze methode worden de buffers van de DataProvider leeggemaakt.

---

### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/buffers/clear>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/buffers/clear?key=YourActivatedAPIKey>

### Antwoord

*Voorbeeld*

```
{
  "status": "done"
}
```

---

## Alle properties

Deze aanvraag haalt alle properties van de applicatie op.

---

### Vraag

*Patroon*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/properties>

*Voorbeeld*

**GET** <http://verkeer-2.bp.tiwi.be/api/v2/settings/properties?key=YourActivatedAPIKey>

### Antwoord

*Voorbeeld*

```
[
  {
    "jndi": "location/of/the/propertyfile",
    "content": {
      "firstvariable": "value"
    }
  },
  {
    "jndi": "location/of/the/propertyfile",
    "content": {
      "firstvariable": "value",
      "secondvariable": "value"
    }
  }
]
```

---

## Update properties

Deze opdracht zal de properties-bestanden aanpassen.

---

### Vraag

*Patroon*

**POST** http://verkeer-2.bp.tiwi.be/api/v2/settings/properties

*Voorbeeld*

**POST** http://verkeer-2.bp.tiwi.be/api/v2/settings/properties?key=**YourActivatedAPIKey**

### Aanvraag

*Voorbeeld*

```
[
  {
    "jndi": "location/of/the/propertyfile",
    "content": {
      "firstvariable": "value"
    }
  },
  {
    "jndi": "location/of/the/propertyfile",
    "content": {
      "firstvariable": "value",
      "secondvariable": "value"
    }
  }
]
```

### Antwoord

*Voorbeeld*

```
{
  "status": "ok"
}
```

---

# Hoofdstuk 3: Kwaliteitscontrole

---

## 3.1 CORE

---

### 3.1.1 Sortering GeoLocations

In één Route zitten meerdere GeoLocations om te verzekeren dat het gevolgde pad correct is. Om zeker te zijn dat de sortering van de locaties correct gebeurt, werd een unittest geschreven.

### 3.1.2 GeoLocation

In de klasse GeoLocation houden twee variabelen de coördinaten *latitude* en *longitude* bij. Deze coördinaten moeten binnen bepaalde grenswaarden liggen. Bijgevolg werd hiervoor een exceptie met bijkomende unittest geschreven.

### 3.1.3 GeoJsonProvider

In de GeoJsonProvider wordt getest of Google reageert op de aanvragen.

### 3.1.4 SourceAdapter

Een adapter zal de data afkomstig van een bepaalde provider aan de hand van een opgegeven route ophalen. Deze data wordt dan in een RouteData-object opgeslagen en via eerder vermelde klassen weggeschreven naar de database. Het testen van het wegschrijven naar de database zelf gebeurt in de TrafficDataDownloader.

De URL en API-Key zijn terug te vinden in het SourceAdapter.properties-bestand, waar ze ook aangepast kunnen worden. Om na te gaan of er een exceptie wordt opgegooid bij een route waar de provider geen data van heeft, is een unittest beschikbaar in de bijhorende SourceAdapter-bean. Bij alle mogelijke problemen dienen excepties gelogd te worden.

De testplannen voor adapters verlopen analoog met enkele uitzonderingen.

#### 1. Ideaal Scenario

- (a) Data wordt opgehaald via URL, API-key en geolocaties van de route
- (b) Afstand en reistijd worden uit de data (JSON-Object) gehaald
- (c) Afstand en reistijd worden in het RouteData-object opgeslagen
- (d) RouteData-object wordt teruggegeven aan de TrafficDataDownloader

## 2. Mogelijke problemen

- (a) URL / API-Key werkt niet meer
  - i. Testen door gebruik te maken van een dummy .properties-bestand
- (b) Internetconnectie werkt niet
  - i. Testen door internet uit te schakelen
- (c) Geen data voor Route beschikbaar (door incorrecte geolocaties)
  - i. Testen door gebruik te maken van de Unittest in de Bean van de desbetreffende SourceAdapter
- (d) Data opgehaald van een verkeerd traject tussen de geolocaties
  - i. Probleem wordt dynamisch opgevangen door TrafficDataDownstreamAnalyser

### 3.1.5 TimerScheduler

Er is een TimerScheduler.properties-bestand aanwezig in de map 'Realisatie' die kan je gebruiken om onderstaande checklist te doorlopen. Hierin pas je de tijdsintervallen aan.

Er wordt een lijn geprint in de log van de server na afloop van een interval. Op deze manier kan de werking van de TimerScheduler gecontroleerd worden. Je neemt ook best een stopwatch bij de hand ter controle.

1. Test met verschillende intervallen
  - (a) Werkt dit voor een interval van 2 minuten?
  - (b) Werkt dit voor een interval van 5 minuten?
  - (c) Werkt dit voor een interval van 10 minuten?
2. Test of tijdens het draaien van de applicatie het interval correct aangepast wordt wanneer gevraagd
  - (a) Wordt de overgang gemaakt van 2 minuten naar 5 minuten?
  - (b) Wordt de overgang gemaakt van 5 minuten naar 10 minuten?
  - (c) Wordt de overgang gemaakt van 10 minuten naar 2 minuten?

### 3.1.6 TrafficDataDownloader - DownstreamAnalyser - DAO

Bij het testen van deze modules wordt gebruik gemaakt van een DummySourceAdapter. Het instellen van de DummySourceAdapter als SourceAdapter gebeurt in het SourceAdapters.properties-bestand. In dit bestand moet de bestaande inhoud vervangen worden door onderstaande inhoud. Het is aan te raden om op voorhand een back-up te nemen van de originele inhoud, anders zal de applicatie daarna niet meer correct werken. Eventueel is het ook mogelijk om deze adapter als extra adapter (met bijhorend volgnummer) toe te voegen in dit bestand.

---

```
1=java:global/Verkeer2/DummySourceAdapter/DummySourceAdapter
```

---

Stel dat de modules getest dienen te worden wanneer de applicatie draait op de server kan bovenstaande inhoud worden toegevoegd aan het originele bestand. Er moet wel rekening gehouden worden met het volgnummer horend bij de adapter, deze moet op elke lijn verschillend zijn.

De data die in DummySourceAdapter gegenereerd wordt ziet eruit zoals in onderstaande tabel.



De teller is dus het volgnummer voor het interval, startend bij nul. De duration wordt aan de hand van de teller berekend.

<i>teller</i>	<i>RouteID</i>	<i>Distance</i>	<i>Duration</i>
0	1	1000	100
1	1	1000	130
2	1	1000	160
3	1	1000	190
4	1	1000	220
5	1	2000	250
6	1	1000	280
7	1	1000	310
8	1	1000	340
9	1	1000	370
0	2	2000	200
1	2	2000	230
2	2	2000	260
3	2	2000	290
4	2	2000	320
5	2	4000	350
6	2	2000	380
7	2	2000	410
8	2	2000	440
9	2	2000	470

In de TimerScheduler wordt bij elk interval een methode in TrafficDataDownloader getriggerd waarbij de data voor alle routes opgehaald wordt voor alle adapters. In dit geval is er dus slechts één adapter. TrafficDataDownstreamAnalyser is een klasse die de data zal controleren en doorgeven aan de TrafficDataDAO die op zijn beurt, de RouteData in de database zal wegschrijven. Er worden twee routes gebruikt voor deze test, deze hebben id 1 en 2.

Bij bovenstaande dummydata zit per Route ook één record (geaccentueerd) die niet in de database mag terechtkomen vanwege een afwijkende afstand. In de TrafficDownstreamAnalyser wordt de afstand gecontroleerd aan de hand van het TrafficDownStreamAnalyser.properties-bestand. Indien dit bestand onderstaande instellingen bevat zal de afstand bij een volgend record maar 10% kleiner of groter mogen zijn dan bij het vorige record.

---

```
isDistanceFilterEnabled=true
distanceDif=1.1
```

---

Indien bij het uitvoeren van deze test bovenstaande data (met uitzondering van de geaccentueerde records) na tien intervallen ook in de database teruggevonden kan worden met correcte timestamp, dan betekent dit dat de data correct doorgegeven wordt van module naar module.

## 3.2 DATA PROVIDER - RESTAPI

---

De DataProvider werd volledig handmatig getest. De data die werd verkregen uit RESTAPI-aanvragen werd vergeleken met de data bij het uitvoeren van een bijhorende query in de database op de server. Zo werden alle methodes in DataProvider getest en werden mogelijke fouten opgespoord.

Zo bleek de recentData-array af en toe leeg te zijn. De oorzaak hiervan was de Blocklist-klasse. Een ander probleem was de snelheid op de trajecten: deze werd op een verkeerde manier berekend.

Een geautomatiseerde testing voor de DataProvider hebben we niet geschreven. Bij nader inzien is dit een grote zwakte. Door het modulaire ontwerp ontdekten we verscheidene problemen die een standaard testframework onmogelijk maakten. Unittests waren niet mogelijk omdat er gebruikt moest gemaakt worden van andere beans die moeten *gedeployed* zijn op de server. Indien we op voorhand het testplan beter hadden uitgewerkt hadden we een testsysteem kunnen implementeren in de applicatie. Na het beseffen van deze zwakte hebben we geopteerd voor handmatige controles.

## 3.3 WEBSITE

---

Het webgedeelte van de applicatie wordt volledig getest via Integration Testing. De website, waarvan de URL terug te vinden is in de handleiding, doet dienst als testomgeving.

### 3.3.1 Startpagina

Op de startpagina zijn er een aantal navigatiemogelijkheden. In bijlage staat een checklist waarin de werking van elke functionaliteit kan worden gecontroleerd.

### 3.3.2 Live

Deze pagina bestaat uit twee delen, namelijk de kaart en de tabel. Op deze pagina zijn er twee mogelijkheden waar hierna dieper wordt op ingegaan: de live-modus zelf en een weergave van de gemiddelden. In bijlage staat een checklist waarin de werking van elke functionaliteit kan worden gecontroleerd.

#### Live - Gemiddeld

Het plan voor het testen van live en gemiddeld is identiek aan elkaar, de checklist voor dit deel moet bijgevolg tweemaal doorlopen worden.

### 3.3.3 Analyse

Er zijn een aantal verschillende analyses die uitgevoerd kunnen worden, deze verlopen grotendeels analoog.

In het stappenplan en in tweede instantie, de sidebar, worden parameters meegegeven aan de analyse. Deze parameters worden pas na het indienen van het formulier gezet in de URL en kunnen bijgevolg ook dan pas nagekeken worden. In bijlage staat een checklist waarin de werking van elke functionaliteit kan worden gecontroleerd.

### 3.3.4 Instellingen

Er zijn verschillende keuzemogelijkheden bij de instellingen. Elk van deze mogelijkheden heeft een andere functionaliteit die gecontroleerd kan worden aan de hand van de bijhorende testplannen in bijlage.

# Hoofdstuk 4: Gebruikshandleiding

---

## 4.1 WEBSITE

---

*URL: <https://http://verkeer-2.bp.tiwi.be/>*

Wanneer je naar de URL van de website surft, kom je terecht op de startpagina. Door op de centrale Login-knop te klikken kom je op de Login-pagina terecht. Hier kan je je door een correct username en password in te vullen aanmelden op de website en zo zal je terechtkomen bij het dashboard van de applicatie. Dit dashboard geeft een overzicht van de mogelijkheden van de applicatie. De applicatie is opgedeeld in 3 grote delen: de live-pagina, de analyse-pagina en de instellingen-pagina. De zijn te bereiken via de tegels op het dashboard. Daarnaast bevat het dashboard ook tegels om snel naar de profiel-pagina en de pagina om trajecten te beheren te gaan. Tenslotte is er ook een link met de API documentatie.

Indien u zich in 1 van de 3 delen van de website bevindt, kan je steeds snel overstappen naar een ander deel via de navigatiebalk bovenaan. Het eerste icoontje verwijst naar het live-gedeelte, het 2e naar de analyses, het 3e naar de instellingen en het 4e brengt je terug naar het dashboard. Daarnaast bevindt er zich nog een knop om de pagina te vernieuwen en onder het laatste icoontje vind je linken om snel naar het profiel, de server logs te gaan of om uit te loggen.

Er zijn ook filmpjes terug te vinden waarin je wegwijs wordt gemaakt op de website, deze zijn terug te vinden op onderstaande URL.

*URL: <https://www.youtube.com/channel/UCfVianoeTHN38Ky9gZxGEmA>*

### 4.1.1 Live-pagina

*URL: <https://verkeer-2.bp.tiwi.be/live>*

De Live-pagina dient om een overzicht te bieden van de actuele verkeerssituatie op alle trajecten, zowel op kaart als in tabelvorm.

Door op het dashboard op de 'huidige situatie'-tegel linksboven te klikken komen we op de livepagina die onderverdeeld is in een kaartgedeelte (links) en een tabelgedeelte (rechts).

Op de kaart kan worden in- en uitgezoomd doormiddel van de + en knop, ofwel door te scrollen. Je kan de kaart ook eenvoudig verslepen.

Door op de kaart op een bepaald traject te klikken, wordt dit traject gehighlight in de tabel. De kleur van het traject op de kaart komt overeen met het kleur van de actuele vertraging en is ook terug te vinden bij de vertraging in de tabel. Groen staat voor normaal verkeer en rood voor grote vertragingen, de legende van de kleuren is linksonder op de kaart terug te vinden.

De tabel aan de rechterkant bevat de volgende zaken voor elk traject:

1. De actuele reistijd in minuten
2. De actuele vertraging ten opzichte van de optimale reistijd in minuten
3. Een pijl die de trend van het verkeer aangeeft (stijgend of dalend)

De tabel is gesorteerd volgens aflopende vertragingen en als je met de muis boven een bepaald traject gaat staan, wordt het overeenkomstig traject op de kaart aangeduid in het zwart.

Boven deze tabel loopt een timer die aangeeft wanneer er nieuwe data beschikbaar zullen zijn. Door rechtsboven op de kaart te klikken op 'gemiddeld' wordt er overgeschakeld op de gemiddelde modus. Deze modus werkt analoog als de livemodus maar geeft de gemiddelde verkeerssituatie voor elk traject terug. In de gemiddelde modus ontbreekt dus wel de pijl die de trend aangeeft. Om terug te keren naar de livemodus kan op live geklikt worden. Door te klikken op het logo linksboven wordt er teruggekeerd naar de startpagina.

## 4.1.2 Analyse-pagina

URL: <https://verkeer-2.bp.tiwi.be/analyse>

### Gemiddelde verkeerssituaties

Deze analyse kan worden gebruikt om de gemiddelde verkeerssituatie op n of meerdere trajecten te bekijken en te vergelijken. De data zullen worden uitgemiddeld over de gewenste bronnen en over de gewenste periode.

Door op het dashboard te kiezen voor 'Analyse' en vervolgens door te klikken op 'Gemiddelde verkeerssituaties', komen we bij het stappenplan om de analyse op te stellen.

1. Eerst kan de periode ingesteld worden waarover de verkeerssituaties zullen worden vergeleken. Stel hiervoor bij 'Begin' en 'Einde' het begin en het einde van de gewenste periode in via de kalender. Zorg ervoor dat de startdatum voor de einddatum komt.

Indien je de periode niet invult, wordt er een standaardperiode gebruikt. Klik vervolgens op 'volgende'.

2. Vervolgens worden de trajecten die je wenst te analyseren aangeduid. Standaard staan alle trajecten geselecteerd. Je kan er enkele deselecteren door erop te klikken of je kan ze allemaal ineens deselecteren door op 'deselecteer alle trajecten' te klikken en vervolgens de gewenste opnieuw te selecteren. Wanneer je over een traject hovert, wordt dit traject aangeduid op de kaart. Als de gewenste traject zijn aangeduid, klikt u op 'volgende'.
3. Tenslotte kan je specificeren welke dataproviders je wenst te selecteren voor het berekenen van de gemiddelde reistijden. Standaard staan alle providers aangeduid maar je kan manueel de providers die niet nodig zijn uitvinken. Klik tenslotte op analyseer.

Er verschijnt nu een tabel waarop elk gekozen traject te vinden is met bijhorende informatie. Deze informatie bestaat uit de afstand, de optimale reistijd in minuten, de optimale snelheid in km/h, de gemiddelde reistijd in minuten, de gemiddelde snelheid in km/h en het percentage verschil tussen optimale en gemiddelde reistijd. Voor dit verschil worden ook kleurcodes gebruikt: groen wijst op een klein verschil, rood op een groot verschil.

U kan ook de aanvraag van deze analyse aanpassen door linksboven op 'wijzig aanvraag' te klikken. In de sidebar die vervolgens verschijnt kan u de start- en einddata van de periode, de gewenste trajecten en/of de nodige dataproviders aanpassen. Hier kan ook de precisie van de aanvraag verhoogd worden. Indien de aanpassingen zijn gebeurd, klikt u op 'analyseren' om de nieuwe tabel te genereren.

## Vertragingen per weekdag

Deze analyse kan worden gebruikt om data (van n of meerdere bronnen) op verschillende dagen (maandag t.e.m. zondag) te vergelijken. De data zal uitgemiddeld worden per dag, dus als in de gekozen periode 4 maandagen zitten zullen deze worden omgezet naar n 'gemiddelde' maandag. Op deze manier kan je dus binnen een bepaalde periode kijken op welke dagen de vertragingen het hoogst zijn.

Door op het dashboard te kiezen voor Analyse en vervolgens door te klikken op 'Vertragingen per weekdag', komen we bij het stappenplan om de analyse op te stellen.

1. Eerst kan de periode ingesteld worden waarover de weekdagen zullen worden vergeleken. Stel hiervoor bij 'Begin' en 'Einde' het begin en het einde van de gewenste periode in via de kalender. Zorg ervoor dat de startdatum voor de einddatum komt.

Indien je de periode niet invult, wordt er een standaardperiode gebruikt. Klik vervolgens op 'volgende'.

2. Vervolgens wordt n traject dat je wenst te analyseren aangeduid in de lijst door erop te klikken. Wanneer je over een traject hovert, wordt dit traject aangeduid op de kaart. Als het gewenste traject is aangeduid, klikt u op 'volgende'.
3. Tenslotte kan je specificeren welke dataproviders je in de berekeningen wenst op te nemen. Standaard staan alle providers aangeduid maar je kan manueel de providers die niet nodig zijn uitvinken. Klik tenslotte op 'analyseer'.

Er verschijnt nu een grafiek waarop elke dag in een verschillend kleur wordt aangeduid. Door te scrollen kan er op de grafiek worden in- en uitgezoomd en wanneer u met de muis op de grafiek gaat staan, krijgt u de exacte data te zien. Door boven de grafiek op 'ingezoomd' te klikken wordt de grafiek meer gecentreerd (de y-as start niet meer bij 0). Dit geeft vaak een duidelijker resultaat.

Indien u een bepaalde weekdag niet meer wenst weer te geven op de grafiek, kan u deze verwijderen door onder de grafiek de desbetreffende weekdag aan te klikken. Door nogmaals op deze weekdag te klikken, zal deze opnieuw op de grafiek verschijnen.

U kan ook de aanvraag van deze analyse aanpassen door linksboven op wijzig aanvraag te klikken. In de sidebar die vervolgens verschijnt kan u de start- en einddata van de periode, het gewenste traject en/of de nodige dataproviders aanpassen. Hier kan ook de precisie van de aanvraag verhoogd worden. Indien de aanpassingen zijn gebeurd, klikt u op 'analyseren' om de nieuwe grafiek te genereren.

## Spitsuren tijdens werkdagen

Deze analyse kan worden gebruikt om de vertragingen op een gemiddelde werkdag (maandag t.e.m. vrijdag) te bekijken. De data zal uitgemiddeld worden op  $n$  dag (een cyclus van 24 uur), dus als in de gekozen periode 10 werkdagen zitten zullen deze worden omgezet naar  $n$  'gemiddelde' dag.

Door op het dashboard te kiezen voor Analyse en vervolgens door te klikken op 'Spitsuren tijdens werkdagen', komen we bij het stappenplan om de analyse op te stellen.

1. Eerst kan de periode ingesteld worden waarover de werkdagen zullen worden geanalyseerd. Stel hiervoor bij 'Begin' en 'Einde' het begin en het einde van de gewenste periode in via de kalender. Zorg ervoor dat de startdatum voor de einddatum komt.

Indien je de periode niet invult, wordt er een standaardperiode gebruikt. Klik vervolgens op 'volgende'.

2. Vervolgens wordt  $n$  traject dat je wenst te analyseren aangeduid in de lijst door erop te klikken. Wanneer je over een traject hoovert, wordt dit traject aangeduid op de kaart. Als het gewenste traject is aangeduid, klikt u op 'volgende'.
3. Tenslotte kan je specificeren welke dataproviders je in de berekeningen wenst op te nemen. Standaard staan alle providers aangeduid maar je kan manueel de providers die niet nodig zijn uitvinken. Klik tenslotte op 'analyseer'.

Er verschijnt nu een grafiek waarop  $n$  lijn te zien is die het verloop van een gemiddelde werkdag weerspiegelt. Op deze manier kan je zien wanneer gemiddeld de meeste vertraging optreedt in de werkweek. Door te scrollen kan er op de grafiek worden in- en uitgezoomd en wanneer u met de muis op de grafiek gaat staan, krijgt u de exacte data te zien.

Door boven de grafiek op 'ingezoomd' te klikken wordt de grafiek meer gecentreerd (de y-as start niet meer bij 0). Dit geeft vaak een duidelijker resultaat.

U kan ook de aanvraag van deze analyse aanpassen door linksboven op wijzig aanvraag te klikken. In de sidebar die vervolgens verschijnt kan u de start- en einddata van de periode, het gewenste traject en/of de nodige dataproviders aanpassen. Hier kan ook de precisie van de aanvraag verhoogd worden. Indien de aanpassingen zijn gebeurd, klikt u op 'analyseren' om de nieuwe grafiek te genereren.

## Trajecten vergelijken

Deze analyse kan worden gebruikt om data (van n of meerdere bronnen) te vergelijken tussen twee of meerdere trajecten. Van de dagen in de periode wordt het gemiddelde genomen zodat het gemiddeld verloop op n dag wordt weergegeven per traject.

Door op het dashboard te kiezen voor Analyse en vervolgens door te klikken op 'Trajecten vergelijken', komen we bij het stappenplan om de analyse op te stellen.

1. Eerst kan de periode ingesteld worden waarover de werkdagen zullen worden geanalyseerd. Stel hiervoor bij 'Begin' en 'Einde' het begin en het einde van de gewenste periode in via de kalender. Zorg ervoor dat de startdatum voor de einddatum komt.

Indien je de periode niet invult, wordt er een standaardperiode gebruikt. Klik vervolgens op 'volgende'.

2. Vervolgens worden de trajecten die je wenst te vergelijken aangeduid in de lijst door erop te klikken. Wanneer je over een traject hooft, wordt dit traject aangeduid op de kaart. Als de gewenste trajecten zijn aangeduid, klikt u op 'volgende'.
3. Tenslotte kan je specificeren welke dataproviders je in de berekeningen wenst op te nemen. Standaard staan alle providers aangeduid maar je kan manueel de providers die niet nodig zijn uitvinken. Klik tenslotte op 'analyseer'.

Er verschijnt nu een grafiek waarop elk traject in een verschillende kleur wordt aangeduid. Door te scrollen kan er op de grafiek worden in- en uitgezoomd en wanneer u met de muis op de grafiek gaat staan, krijgt u de exacte data te zien. Door boven de grafiek op 'ingezoomd' te klikken wordt de grafiek meer gecentreerd (de y-as start niet meer bij 0). Dit geeft vaak een duidelijker resultaat.

Indien u een bepaald traject niet meer wenst weer te geven op de grafiek, kan u deze verwijderen door onder de grafiek het desbetreffende traject aan te klikken. Door nogmaals op dit traject te klikken, zal dit traject opnieuw op de grafiek verschijnen.

U kan ook de aanvraag van deze analyse aanpassen door linksboven op 'wijzig aanvraag' te klikken. In de sidebar die vervolgens verschijnt kan u de start- en einddata van de periode, de gewenste trajecten en/of de nodige dataproviders aanpassen. Hier kan ook de precisie van de aanvraag verhoogd worden. Indien de aanpassingen zijn gebeurd, klikt u op 'analyseren' om de nieuwe grafiek te genereren.



## Perioden vergelijken

Deze analyse kan worden gebruikt om data (van n of meerdere bronnen) te vergelijken tussen twee of meerdere periodes.

Door op het dashboard te kiezen voor Analyse en vervolgens door te klikken op 'Perioden vergelijken', komen we bij het stappenplan om de analyse op te stellen.

1. Eerst moeten de periodes ingesteld worden die met elkaar zullen worden vergeleken. Klik hiervoor op Nieuwe Periode en vul met behulp van de kalender de start- en einddatum van de periode in. Zorg ervoor dat de startdatum voor de einddatum komt en klik vervolgens op voeg toe. Doe dit voor elk van de te vergelijken periodes. Als alle periodes correct zijn toegevoegd, klikt u op 'volgende'.
2. Vervolgens wordt n traject dat je wenst te analyseren aangeduid in de lijst door erop te klikken. Wanneer je over een traject hooft, wordt dit traject aangeduid op de kaart. Als het gewenste traject is aangeduid, klikt u op 'volgende'.
3. Tenslotte kan je specificeren welke dataproviders je in de berekeningen wenst op te nemen. Standaard staan alle providers aangeduid maar je kan manueel de providers die niet nodig zijn uitvinken. Klik tenslotte op 'analyseer'.

Er verschijnt nu een grafiek waarop elke periode in een verschillend kleur wordt aangeduid. Door te scrollen kan er op de grafiek worden in- en uitgezoomd en wanneer u met de muis op de grafiek gaat staan, krijgt u de exacte data te zien. Door boven de grafiek op 'ingezoomd' te klikken wordt de grafiek meer gecentreerd (de y-as start niet meer bij 0). Dit geeft vaak een duidelijker resultaat.

Indien u een bepaalde periode niet meer wenst weer te geven op de grafiek, kan u deze verwijderen door onder de grafiek de desbetreffende periode aan te klikken. Door nogmaals op deze periode te klikken, zal deze periode opnieuw op de grafiek verschijnen.

Door rechtsboven op tabel te klikken, kunnen we de data van deze analyse in tabelvorm bekijken. Door op grafiek te klikken keer je terug naar de grafiek.

U kan ook de aanvraag van deze analyse aanpassen door linksboven op wijzig aanvraag te klikken. In de sidebar die vervolgens verschijnt kan u een nieuwe periode toevoegen en het gewenste traject en/of de nodige dataproviders aanpassen. Hier kan ook de precisie van de aanvraag verhoogd worden. Indien de aanpassingen zijn gebeurd, klikt u op 'analyseren' om de nieuwe grafiek te genereren.

## Databronnen vergelijken

Deze analyse kan gebruikt worden om de data van twee of meerdere databronnen te vergelijken.

Door op het dashboard te kiezen voor analyse en vervolgens door te klikken op 'databronnen vergelijken', komen we bij het stappenplan om de analyse op te stellen.

1. Eerst kan de periode ingesteld worden waarover de databronnen zullen worden geanalyseerd. Stel hiervoor bij 'Begin' en 'Einde' het begin en het einde van de gewenste periode in via de kalender. Zorg ervoor dat de startdatum voor de einddatum komt.

Indien je de periode niet invult, wordt er een standaardperiode gebruikt. Klik vervolgens op 'volgende'.

2. Vervolgens wordt n traject dat je wenst te analyseren aangeduid in de lijst door erop te klikken. Wanneer je over een traject hooft, wordt dit traject aangeduid op de kaart. Als het gewenste traject is aangeduid, klikt u op 'volgende'.
3. Tenslotte kan je specificeren welke dataproviders je wenst te vergelijken met elkaar. Standaard staan alle providers aangeduid maar je kan manueel de providers die niet nodig zijn uitvinken. Klik tenslotte op 'analyseer'.

Er verschijnt nu een grafiek waarop elke dataprovider in een verschillend kleur wordt aangeduid. Door te scrollen kan er op de grafiek worden in- en uitgezoomd en wanneer u met de muis op de grafiek gaat staan, krijgt u de exacte data te zien. Door boven de grafiek op 'ingezoomd' te klikken wordt de grafiek meer gecentreerd (de y-as start niet meer bij 0). Dit geeft vaak een duidelijker resultaat. Indien u een bepaalde dataprovider niet meer wenst weer te geven op de grafiek, kan u deze verwijderen door onder de grafiek de desbetreffende provider aan te klikken. Door nogmaals op deze provider te klikken, zal deze provider opnieuw op de grafiek verschijnen.

Door rechtsboven op tabel te klikken, kunnen we de data van deze analyse in tabelvorm bekijken. Door op grafiek te klikken keer je terug naar de grafiek.

U kan ook de aanvraag van deze analyse aanpassen door linksboven op wijzig aanvraag te klikken. In de sidebar die vervolgens verschijnt kan u de start- en einddata van de periode, de gewenste trajecten en/of de nodige dataproviders aanpassen. Hier kan ook de precisie van de aanvraag verhoogd worden. Indien de aanpassingen zijn gebeurd, klikt u op 'analyseren' om de nieuwe grafiek te genereren.

### 4.1.3 Instellingen-pagina

URL: <https://verkeer-2.bp.tiwi.be/settings>

De settingspagina dient om makkelijk wijzingen aan te kunnen brengen in de applicatie. Deze pagina kan bereikt worden door op het dashboard op 'Instellingen' te klikken. Vervolgens krijg je een overzicht te zien van de verschillende instellingen die kunnen gebeuren. In wat volgt worden deze kort overlopen.

#### Trajecten beheren

Via deze pagina, die ook rechtstreeks via het dashboard te bereiken is, krijg je een overzicht van alle aanwezige trajecten en is het mogelijk om de trajecten te beheren.

Zo kan er een nieuw traject worden toegevoegd door bovenaan rechts op het plusteken te klikken. Vervolgens vul je de gegevens van het nieuwe traject in, te beginnen met een naam. Daarna geef je bij 'van' en 'naar' de namen van de begin- en eindlocatie mee en duid je deze punten aan op een kaart zodat de correcte geolocaties aan dit punt worden toegevoegd. Door op 'toevoegen' te klikken wordt het traject effectief aan de lijst toegevoegd.

Elk traject uit de lijst kan ook verwijderd worden door op het kruisje naast het te verwijderen traject te klikken.

Door op het eerste icoontje (potlood) naast een traject in de lijst te klikken, kan je dit traject bewerken. Zo kan je de naam aanpassen of het aantal seconden dat bij elke threshold van dit traject hoort verhogen of verlagen. De wijzigingen worden opgeslagen als je op 'wijzig' klikt.

Wanneer je op een traject uit de lijst klikt of op het tweede icoontje naast een traject, kom je op een pagina waar je geolocaties en/of handlers aan het traject kan toevoegen of deze juist verwijderen. Om een geolocatie toe te voegen klik je op 'voeg tussenpunt toe'. Vervolgens geef je dit punt een naam en duid je de correcte plaats aan op de kaart zodat de geolocaties correct worden toegevoegd. Tenslotte geef je aan na welk tussenpunt dit nieuw tussenpunt komt door bij 'invoegen na' het voorgaande tussenpunt uit de lijst te selecteren. Klik vervolgens op 'toevoegen' en het traject zal een nieuw tussenpunt bevatten.

Tussenspuncten kunnen verwijderd worden door op het kruisje naast het tussenpunt in de lijst te klikken.

Je kan handlers, zoals bijvoorbeeld Twitter, aan een bepaalde threshold toevoegen. Als deze threshold overschreden wordt, zal dit de gekoppelde handler activeren om dan bijvoorbeeld een melding op Twitter te zetten. Klik hiervoor op 'voeg nieuwe melding toe'. Daarna selecteer je bij 'melding' welke handler (op dit moment is dit enkel Twitter) je wil koppelen en geef je aan bij 'vanaf vertraging' aan welke threshold je deze handler wil koppelen. Tenslotte klik je op 'toevoegen'.

Door op 'reset meldingen' te klikken worden alle gekoppelde handlers weer verwijderd.

#### Systeem

Op deze pagina krijg je een overzicht van alle standaardinstellingen van de applicatie die normaal in een properties-bestand moeten worden aangepast maar hier dus gemakkelijk kunnen worden aangepast.

Nadat de gewenste instellingen zijn aangepast kunnen deze doorgevoerd worden door op 'wijzig' te klikken.

## Logs Server

Hier vind je een overzicht terug van de laatste 100 logs die op de server geregistreerd zijn met de recentste bovenaan.

## Gebruikers

Op deze pagina vind je een lijst terug met de naam en gebruikersnaam van alle gebruikers. Je kan hier ook een nieuwe gebruiker toevoegen door onderaan 'naam', 'gebruikersnaam' en 'password' in te vullen en vervolgens op 'genereer nieuwe gebruiker' te klikken.

## Profiel

Hier vind je de gegevens van het profiel waarmee je ingelogd bent terug. Je kan deze gegevens ook aanpassen en deze aanpassingen vervolgens opslaan door op 'toevoegen' te klikken.

## API-Keys

Via deze pagina krijg je een lijst van alle (in)actieve API-Keys die nodig zijn om toegang te krijgen tot de beveiligde calls van de API. Je kan elke sleutel deactiveren door op 'deactiveren' naast de sleutel in de lijst te klikken, deze sleutel kan dan niet meer gebruikt worden. Door onderaan op 'genereer nieuwe sleutel' te klikken, wordt er een nieuwe, actieve sleutel aan de lijst toegevoegd.

### 4.1.4 API

*URL: <http://docs.verkeerscentrumgent.apiary.io/>*

Op de startpagina kan je doorklikken naar de API, deze zal je doorverwijzen naar de documentatie op apiary horend bij deze applicatie. Meer informatie hierover is terug te vinden in de handleiding met betrekking tot de API.

## 4.2 API

---

*URL: <http://docs.verkeerscentrumgent.apiary.io/>*

De REST API geeft info terug uit de database, hier heb je de keuze uit verschillende opties die in de volgende delen besproken worden. Wanneer je hier aanvragen naar doet krijg je gegevens uit de database terug. In het data ontwerp wordt een korte toelichting gegeven per optie, gevolgd door een voorbeeld van aanvraag en antwoord in JSON-formaat. Deze documentatie kan echter ook allemaal teruggevonden worden op bovenstaande URL.

# Hoofdstuk 5: Installatiehandleiding

---

## 5.1 SYSTEEMEISEN

---

### 5.1.1 Systeemvereisten

1. 1GB memory minimum (2GB memory recommended)
2. 500MB disk space minimum (1GB disk space recommended)
3. Supported platforms: Solaris 10, OpenSolaris 2009, Red hat enterprise linux 4+, Ubuntu linux 8+, Windows xp SP3+, Mac OS X 10.5+
4. JDK 7 minimum

### 5.1.2 Software

1. GlassFish Server Open Source Edition 4.1 (build 13) - Java ee container voor de werkelijke applicatie
2. MySQL server (Project tested on Linux Ubuntu with '5.5.47-0ubuntu0.14.04.1 (Ubuntu mariadb' ) - Database voor opslag data
3. *(optioneel) nginx - Instellen van forwarding naar http / https poorten van de bovenstaande applicaties*

### 5.1.3 Software

De gevraagde applicatie is geprogrammeerd in Java, dit laat toe om de applicatie op alle besturingssystemen te draaien zolang deze Java ondersteunen. Er zijn dan ook geen eisen aan de hardware voor deze applicatie, enkel softwarepakketten zijn vereist voor het draaien van deze software.

Er is steeds een actieve internetverbinding vereist om de data van de providers te ontvangen.

#### Geteste software

1. Java jre1.8.0\_73 of Java jdk1.8.0\_73
2. Glassfish 4.1 (build 13)
3. MySQL Server 5.7
4. MySQL workbench 6.3 CE (aanbevolen voor weergave databank)
5. NetBeans 8.1 full installation

## 5.2 INSTELLEN SYSTEEM

---

### Setup/Installatiemap

Dit project is een modulaair systeem bestaande uit tientallen verschillende java enterprise beans. Om deze allen aan te bieden in 1 geheel, alsook het management hiervan is in de distribution map een verzameling aangelegd. In het vervolg van deze tekst zal hiernaar vermeld worden als 'installatiemap' of 'setupmap'.

De bash shell scripts aanwezig in deze map kunnen gebruikt worden indien het project op een linuxdistributie is geïnstalleerd. Hierbij dient wel op voorhand de 'configuration.sh' te worden aangepast naar het systeem waarop deze installatie is gedeployed. Vergeet hierbij vooral niet bij 'glassfishInstall' naar de installatielocatie van glassfish te verwijzen en bij 'verkeer2Install' naar het path van de installatiemap te verwijzen.

### Database

De MySQL database dient reeds gestart te zijn alvorens dit aan te vatten. Afhankelijk van de geïnstalleerde versie en besturingssysteem kan het start-stop commando verschillen. Contacteer uw systeemadministrator voor meer info en login gegevens.

Enkele voorbeelden voor linux: `systemctl start mariadb.service`, `systemctl start mysql`, `/etc/init.d/mariadb.service start`, `/etc/init.d/mysql start`.

Login op de MySQL service.

Onder linux: `mysql -user=user_name -password=your_password`

De database dient gecreëerd te worden samen met een gebruiker die door de applicatie zal worden gebruikt.

Dit script is aanwezig in de setup package onder de naam 'setup.sql' en kan via fileinput uitgevoerd worden in de MySQL service.

Onder linux: `mysql -user=user_name -password=your_password <- setup.sql`

Voor een lokale database met gebruikersaccount 'verkeer' en paswoord 'vop2016' volstaan volgende commando's:

```
CREATE DATABASE verkeer2;
USE verkeer2;
CREATE USER 'verkeer'@'localhost' IDENTIFIED BY 'vop2016';
GRANT ALL ON verkeer2.* TO 'verkeer'@'localhost';
```

## Glassfish

Eens de database is ingesteld kunnen alle resources voor Glassfish worden aangemaakt. Hieronder staat omschreven welke acties dienen uitgevoerd te worden.

Onderstaand proces is geautomatiseerd voor linux en het bash bestand 'setup.sh' is terug te vinden in het setup package.

Login op de 'Admin console interface'. Consulteer uw administrator om het poortnummer en de logingegevens te verkrijgen. Voor de default poort bevindt zich de interface op `http://server.adres:4848`.

1. Voer allereerst alle *fixes* door in Glassfish
2. Voeg de externe *libraries* toe aan Glassfish
3. Herstart Glassfish
4. Voeg JNDI *resources* toe
5. Voeg JDBC connectie toe

Afhankelijk van een lokale testinstallatie van Netbeans of een server dient een iets ander stappenplan te worden gevolgd:

### *Lokale server*

Verifieer dat de Beans en SourceAdapter property-bestanden de applicatiennaam voor hun JNDI naam hebben staan. Deze moeten voldoen aan volgend formaat: `java:global/ApplicatieNaam/ContainerNaam/BeanNaam!remoteOrLocalInterface`.

voorbeeld= `DataProvider=java:global/Verkeer2/DataProvider/DataProvider!iii.vop2016.verkeer2.ejb.dataprovider.DataProviderLocal`

### *Externe server (hier wordt iedere bean apart gedeployed)*

Verifieer dat de Beans en SourceAdapter property-bestanden de applicatiennaam niet voor hun JNDI naam hebben staan. Deze moeten voldoen aan volgend formaat: `java:global/ContainerNaam/BeanNaam!remoteOrLocalInterface`.

voorbeeld= `DataProvider=java:global/DataProvider/DataProvider!iii.vop2016.verkeer2.ejb.dataprovider.DataProviderLocal`

## 5.2.1 Inladen modules

Gebruik als laatste het `./cycle.sh` script om alle beans in te laden in Glassfish en het systeem op te starten voor gebruik. Let hierbij niet op de output van de eerste commando's, Dit script dient eigenlijk om een draaiende server volledig opnieuw op te starten.

Via de admin console van Glassfish kunnen alle modules worden ingeladen. Dit kan onder de tab 'Applications'

Login op de 'Admin console interface'. Consulteer uw administrator om het poortnummer en logingegevens te verkrijgen. Voor de default poort bevindt zich de interface op `http://server.adres:4848`.

Onder de tab 'Applications' kan je module per module toevoegen dia het 'deploy' commando. Zorg ervoor dat steeds alle eigenschappen voor de bean zijn ingevuld. Het is aangeraden het linux bash script `./cycle.sh` te bekijken om alle eigenschappen te zien van iedere bean.



## 5.3 BEHEER

---

### 5.3.1 Start/Stop

Er zijn 2 scripts aanwezig om de services te starten/stoppen: `./start.sh` start Glassfish, `./stop.sh` stopt de service.

### 5.3.2 Logs bekijken

`./viewlog.sh` heeft een link die de Glassfish logs opent via een *append*-mechanisme. Je krijgt live de oude en nieuwere logs te zien tot het script wordt onderbroken.

### 5.3.3 Aanpassen systeem

Via `./deploy.sh` kan je modules die niet geladen zijn inladen. Indien je een nieuwe module zelf geschreven hebt, kan je deze toevoegen aan het script door de vorige modules in het script te analyseren. Zolang de core 'VerkeersLib' van de applicatie niet is veranderd kan dit zonder de server te herstarten.

Via `./redeploy.sh` kan je een geladen module opnieuw inladen. Dit kan gebruikt worden indien een ontwikkelaar een module heeft gewijzigd en wilt toepassen op de server terwijl deze nog draaiende is.

Via `./undeploy.sh` kan je een module terug uitladen. Het wordt ten sterkste afgeraden om een uitgeladen module terug in te laden via `deploy`. Als resources verbonden waren aan deze bean bestaat de kans dat de nieuwe deze resources niet kan alloceren.

Alle modules bevinden zich in de submap `jars/verkeer2/` van de installatiemap. Deze kunnen hier worden gewijzigd en vervolgens opnieuw ingeladen via bovenstaande commando's.

Via `./cycle.sh` kan je alle modules uitladen en opnieuw laten inladen. Deze methode stopt de server en start deze opnieuw op. De functie wordt vooral gebruikt bij aanpassingen als de core 'Verkeerlib' van de applicatie is gewijzigd en moet worden doorgevoegd naar alle modules.

Als men via netbeans of dergelijke aan debugging wenst te doen kan men de server ook opstarten via `./debug.sh`

## 5.4 *Fixes* IN GLASSFISH

---

Er dienen een aantal fixes uitgevoerd te worden in de *admin console* van de Glassfish Server.

Op deze pagina ga je naar Configurations - server.config - JVM Settings, daar selecteer je JVM Options. Hier dienen volgende JVM Options toegevoegd te worden:

```
-Dorg.omg.CORBA.ORBSingletonClass=com.sun.corba.se.impl.orb.  
ORBSingleton
```

Deze fix is nodig sinds JDK 1.7u25 wegens het afwezig zijn van het `com.sun.corba.ee-package`. Deze kan vervangen worden door `com.sun.corba.se`.

```
-Duser.timezone=Europe/Brussels
```

Dit is nodig om de tijdszone van de applicatie in België te laten staan, stel dat de applicatie op een server in een andere tijdszone draait dan zal toch nog de Belgische tijdzone gebruikt worden.

Verder dient ook volgend commando uitgevoerd te worden in de map waar Glassfish geïnstalleerd staat.

```
asadmin set server.ejb-container.property.disable-nonportable-jndi-names="true"
```

Deze fix is nodig om de v2 Vendor-Specific JNDI Names te *disablen*. Zonder deze fix kunnen 2 EJB's niet dezelfde *remote interface* hebben omdat hun JNDI naamgeving overlapt.

## 5.5 EXTERNE *libraries* IN GLASSFISH

---

Alle *libraries* die in de `libs-map` van het project staan moeten aan Glassfish toegevoegd worden. Verder dient ook nog de Verkeerlib (terug te vinden in Realisatie\Verkeerlib\dist) toegevoegd te worden, deze kan je terugvinden in de Verkeerlib-map. Het toevoegen van de *libraries* in Glassfish doe je door de bestanden te plaatsen in volgende twee mappen binnen de Glassfish-map.

1. `/glassfish/lib`
2. `/glassfish/domains/domain1/lib/applibs`

## 5.6 GLASSFISH RESOURCES

---

Hieronder zijn alle resources vermeld die worden gebruikt. Deze moet manueel worden toegevoegd in het administratorbeheer van Glassfish. In onderstaande instructies wordt vaak de term ABSOLUTE\_LOCATION\_TO\_PROJECT gebruikt, dit is de locatie van het project op de server.

### Beans

JNDI Name: resources/properties/Beans  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/Beans.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/Properties.properties

### Properties

JNDI Name: resources/properties/Properties  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/WazeMapping.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/Properties.properties

### WazeMapping

JNDI Name: resources/properties/WazeMapping  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/WazeMapping.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/WazeMapping.properties

### CoyoteMapping

JNDI Name: resources/properties/CoyoteMapping  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/CoyoteMapping.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/CoyoteMapping.properties

### TrafficDataDownstreamAnalyser

JNDI Name: resources/properties/TrafficDataDownstreamAnalyser  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/TrafficDataDownstreamAnalyser.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/TrafficDataDownstreamAnalyser.properties

### Logger

JNDI Name: resources/properties/Logger  
Resource Type: java.util.Properties  
Properties:

```
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/Logger.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/Logger.properties
```

### **SourceAdaptersKeys**

```
JNDI Name: resources/properties/SourceAdaptersKeys
Resource Type: java.util.Properties
Properties:
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/SourceAdaptersKeys.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/SourceAdaptersKeys.properties
```

### **Twitter**

```
JNDI Name: resources/properties/Twitter
Resource Type: java.util.Properties
Properties:
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/Twitter.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/Twitter.properties
```

### **ThresholdHandlers**

```
JNDI Name: resources/properties/ThresholdHandlers
Resource Type: java.util.Properties
Properties:
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/ThresholdHandlers.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/ThresholdHandlers.properties
```

### **DataProvider**

```
JNDI Name: resources/properties/DataProvider
Resource Type: java.util.Properties
Properties:
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/DataProvider.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/DataProvider.properties
```

### **TimerScheduler**

```
JNDI Name: resources/properties/TimerScheduler
Resource Type: java.util.Properties
Properties:
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/TimerScheduler.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/TimerScheduler.properties
```

### **ThresholdManager**

```
JNDI Name: resources/properties/ThresholdManager
Resource Type: java.util.Properties
Properties:
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=
ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/ThresholdManager.properties
propertyLocation=ABSOLUTE_LOCATION_TO_PROJECT/Realisatie/ThresholdManager.properties
```

### GeoJsonProvider

JNDI Name: resources/properties/GeoJsonProvider  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/GeoJsonProvider.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/GeoJsonProvider.properties

### SourceAdapters

JNDI Name: resources/properties/SourceAdapters  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/SourceAdapters.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/SourceAdapters.properties

### WebSettings

JNDI Name: resources/properties/WebSettings  
Resource Type: java.util.Properties  
Properties:  
org.glassfish.resources.custom.factory.PropertiesFactory.fileName=  
ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/WebSettings.properties  
propertyLocation=ABSOLUTE\_LOCATION\_TO\_PROJECT/Realisatie/WebSettings.properties

## 5.7 GLASSFISH JDBC

---

Hieronder staat de JDBC connection pool die gebruikt wordt in de applicatie. Deze moet manueel worden toegevoegd in het administratorbeheer van Glassfish.

### Connection Pool

Pool Name: generalMysql  
Resource Type: javax.sql.DataSource  
Database Driver Vendor: MySQL  
  
additional properties  
password: \*\*\*\*  
databaseName: verkeer2  
serverName: 127.0.0.1  
user: \*\*\*\*  
portNumber: 3306  
url: jdbc:mysql://localhost:3306/verkeer2

### Resources

JNDI Name: jdbc/verkeer/general  
Pool Name: generalMysql

## Hoofdstuk 6: Conclusie

---

Bij de start van het project hebben we nagedacht over wat de meest ideale structuur zou zijn voor de applicatie. We kwamen al snel op het idee om Java EE te gebruiken met de daarbijhorende modulaire beans. Er werden een aantal databronnen aangeboden waarvan data moest worden opgehaald. Door de modulariteit is het mogelijk om later, wanneer de applicatie al draait op de server, extra databronnen toe te voegen zonder dat de applicatie offline wordt gehaald. Dit geldt ook voor het toevoegen van handlers (Twitter).

In de eerste sprint was het doel om de basisstructuur van de applicatie zo goed mogelijk op te bouwen. Er werd bijgevolg voor gekozen om de databank, RESTAPI en een paar SourceAdapters te implementeren. In de tweede sprint werd de RESTAPI herwerkt en uitgebreid samen met de DataProvider. Eveneens werd de webapplicatie opgebouwd. Omdat we graag een zo uitgebreid mogelijke applicatie hadden werd ook de derde sprint nog gevuld met een groot aantal *issues*. In de derde sprint werden de overige analyses in de webapplicatie toegevoegd. Er kwam eveneens een extra websectie waarop allerlei zaken voor de applicatie kunnen ingesteld worden.

Er werden na afronding van het project nog een aantal problemen of mogelijke verbeteringen ontdekt. Deze worden hieronder opgelijst.

1. Geen periode meegeven bij Databronnen vergelijken: vreemd resultaat
2. In de live-modus is het niet mogelijk om snel onderscheid te maken tussen de verschillende providers. Dit is echter wel mogelijk in de analyse-modus, wat wel extra klikwerk met zich meebrengt.
3. Het bepalen van de oorzaak van de vertraging hebben we niet geïmplementeerd vanwege tijdsgebrek.
4. Er is geen onderscheid gemaakt tussen een publieke gebruiker en een medewerker van het verkeerscentrum. Bijvoorbeeld de live-modus zou ook beschikbaar mogen zijn voor de inwoners van Gent.
5. Op elke analyse-pagina is het niet mogelijk om de datums te wijzigen, dit vanwege een foutieve omzetting van het datumformaat.
6. Het is niet mogelijk de trend van de laatste vijftien minuten weer te geven op de live-modus.
7. Indien bij Databronnen vergelijken geen periode wordt er foutieve informatie weergegeven, dit is echter geen probleem wanneer er handmatig een periode wordt gedefinieerd.
8. De beveiliging van de hele applicatie is eenvoudig opgebouwd.
9. Een geautomatiseerde testing voor de DataProvider hebben we niet geschreven. Bij nader inzien is dit een grote zwakte. Door het modulaire ontwerp ontdekten we verscheidene problemen die een standaard testframework onmogelijk maakten. Unittests waren niet mogelijk omdat er gebruikt moest gemaakt worden van andere beans die moeten *gedeployed* zijn op de server. Indien we op voorhand het testplan beter hadden uitgewerkt hadden we een testsysteem kunnen implementeren in de applicatie. Na het beseffen van deze zwakte hebben we geopteerd voor handmatige controles.

# Hoofdstuk 7: Taakverdeling

---

## 7.1 MIKE BRANTS

---

1. Aanmaak DAO's voor database
2. Basis RESTAPI
3. Website

## 7.2 THOMAS VANDE WEGHE

---

1. SourceAdapter: GoogleMaps
2. RESTAPI (implementatie + documentatie)

## 7.3 TOBIAS VAN DER PULST

---

1. Uitwerken framework Java EE, concept modulariteit, *hot-pluggable* principe
2. SourceAdapter: Waze
3. Installatiehandleiding
4. DataProvider, DownstreamAnalyser, Timerscheduler, General/TrafficDataDAO, Thresholds, TwitterHandler, GeoJsonProvider, Settings, Logger
5. Beheer server

## 7.4 SIMON VERMEERSCH

---

1. SourceAdapters: Here, TomTom, Coyote
2. Website: mockups, grafieken
3. Beheer/afwerking verslag
4. Testing

# Hoofdstuk 8: Bijlage

---

## 8.1 TESTPLAN WEBSITE

---

Hieronder vindt u een aantal lijsten terug die gebruikt kunnen worden om de verschillende secties op de webpagina te testen. In de laatste kolom kan de verschillende vragen afgevinkt worden.

### 8.1.1 Startpagina

Live	
Wordt bij hovering de knop geaccentueerd?	
Word je bij klikken doorverwezen naar de Live-pagina?	
Analyses	
Wordt bij hovering de knop geaccentueerd?	
Word je bij klikken doorverwezen naar de Analyses-pagina?	
Instellingen	
Wordt bij hovering de knop geaccentueerd?	
Word je bij klikken doorverwezen naar de Instellingen-pagina?	
API	
Wordt bij hovering de knop geaccentueerd?	
Word je bij klikken doorverwezen naar een pagina met documentatie over de API?	

### 8.1.2 Live

<i>Switch</i> -knop tussen Live en Gemiddeld	
van Live naar Gemiddeld	
Wordt de gemiddelde data in de tabel weergegeven?	
Wordt de gemiddelde data op de kaart weergegeven?	
van Gemiddeld naar Live	
Wordt de live data in de tabel weergegeven?	
Wordt de live data op de kaart weergegeven?	
Keuzemogelijkheden in balk bovenaan	
Links	
Ga je bij het klikken op het logo naar de indexpagina?	
Rechts (knoppen overlopen van links naar rechts)	
Ga je bij het klikken op de eerste knop naar de Live-pagina?	
Ga je bij het klikken op de tweede knop naar de Analyse-pagina?	
Ga je bij het klikken op de derde knop naar de Instellingen-pagina?	
Ga je bij het klikken op de vierde knop naar de startpagina?	
Wordt bij het klikken op de vijfde knop de pagina vernieuwd?	
Ga je bij het klikken op de zesde knop naar het inlogscherf?	



## Live-Gemiddeld

Switch rechtsbovenaan tabel	
Routes met vertraging	
Worden enkel routes met vertraging getoond op de kaart?	
Worden enkel routes met vertraging getoond in de tabel?	
Alle routes	
Worden alle routes getoond op de kaart?	
Worden alle routes getoond in de tabel?	
Kaart	
Zoomfunctie	
Werkt inzoomend scrollen?	
Werkt uitzoomend scrollen?	
Wordt er ingezoomd bij gebruik van de '+'-knop?	
Wordt er uitgezoomd bij gebruik van de '-'-knop?	
Routes	
Standaard	
Worden alle routes correct en met richting op kaart weergegeven? (document met alle routes zal meegeleverd worden)	
Hebben alle routes de correcte kleur (afhankelijk van de threshold)?	
Functionaliteit	
Hover over een route	
Wordt (indien nodig) gescrolld naar de juiste route in de tabel?	
Wordt de route geaccentueerd in de tabel?	
Verdwijnt de accentuering na stoppen met hover?	
Klik op een route	
Verschijnt er een pop-up scherm?	
Bevat de pop-up info van het laatste kwartier in een grafiek?	
Er is een pijl die de trend correct aangeeft?	
Het pop-up scherm verdwijnt na klikken op het kruisje?	
Verschijnt er andere informatie bij klikken op een andere route?	
Tabel	
Standaard	
Wordt bij elke route correcte reistijd weergegeven?	
Wordt bij elke route correcte vertraging weergegeven?	
Is er een scrollbar indien niet alle routes op één pagina passen?	
Verschijnt er relevante informatie in de pop-ups rechtsonderin?	
Wordt het interval in de balk bovenin correct weergegeven?	
Wordt de data gepdate na afloop van het interval?	
Functionaliteit	
Hover over een route	
Wordt de route geaccentueerd in de tabel?	
Verdwijnt de accentuering na stoppen met hover?	
Klik op reistijd	
Wordt de tabel gesorteerd van grote naar kleine reistijd?	
2e klik: wordt de tabel omgekeerd gesorteerd?	
Klik op vertraging	
Wordt de tabel gesorteerd van grote naar kleine vertraging?	
2e klik: wordt de tabel omgekeerd gesorteerd?	

### 8.1.3 Analyse

#### Gemiddelde verkeerssituatie

Stappenplan	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Tijdens deze analyse klik je op een andere analyse	
Worden (indien mogelijk) de parameters behouden?	
Grafiek	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Tabel	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	
Sidebar	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Klik op analyseer	
Worden de parameters in de URL aangepast?	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	

## Vertragingen per weekdag

Stappenplan	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provicer te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Tijdens deze analyse klik je op een andere analyse	
Worden (indien mogelijk) de parameters behouden?	
Grafiek	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Tabel	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	
Sidebar	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provier selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Klik op analyseer	
Worden de parameters in de URL aangepast?	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	

## Spitsuren tijdens werkdagen

Stappenplan	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Tijdens deze analyse klik je op een andere analyse	
Worden (indien mogelijk) de parameters behouden?	
Grafiek	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Tabel	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	
Sidebar	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Klik op analyseer	
Worden de parameters in de URL aangepast?	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	

## Trajecten vergelijken

Stappenplan	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht minimaal twee routes te selecteren?	
Is het zo dat als je geen twee routes selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de Route-ID's in de URL?	
Provider kiezen	
Word je verplicht minimaal één provicer te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Tijdens deze analyse klik je op een andere analyse	
Worden (indien mogelijk) de parameters behouden?	
Grafiek	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Tabel	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	
Sidebar	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht minimaal twee routes te selecteren?	
Is het zo dat als je twee routes selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de Route-ID's in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provier selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Klik op analyseer	
Worden de parameters in de URL aangepast?	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	

## Perioden vergelijken

Stappenplan	
Periode instellen	
Verschijnen na afloop de startdata in de URL?	
Verschijnen na afloop de einddata in de URL?	
Word je verplicht einddata later dan startdata in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Tijdens deze analyse klik je op een andere analyse	
Worden (indien mogelijk) de parameters behouden?	
Grafiek	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Tabel	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	
Sidebar	
Periode instellen	
Verschijnen na afloop de startdata in de URL?	
Verschijnen na afloop de einddata in de URL?	
Word je verplicht einddata later dan de startdata in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provier selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Klik op analyseer	
Worden de parameters in de URL aangepast?	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	

## Databronnen vergelijken

Stappenplan	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Tijdens deze analyse klik je op een andere analyse	
Worden (indien mogelijk) de parameters behouden?	
Grafiek	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Tabel	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	
Sidebar	
Periode instellen	
Verschijnt na afloop de startdatum in de URL?	
Verschijnt na afloop de einddatum in de URL?	
Word je verplicht een einddatum later dan de startdatum in te stellen?	
Route kiezen	
Word je verplicht maximaal één route te selecteren?	
Is het zo dat als je geen route selecteert, de knop 'Volgende' niet werkt?	
Verschijnt na afloop de Route-ID in de URL?	
Provider kiezen	
Word je verplicht minimaal één provider te selecteren?	
Is het zo dat als je geen provider selecteert, de knop 'Volgende' niet werkt?	
Verschijnen na afloop de providers in de URL?	
Klik op analyseer	
Worden de parameters in de URL aangepast?	
Wordt de correcte data weergegeven in de grafiek (nakijken met RESTAPI)?	
Wordt de correcte data weergegeven in de tabel (nakijken met RESTAPI)?	

## 8.1.4 Instellingen

### Trajecten beheren

<b>Overzicht</b>	
Wordt er een lijst weergegeven met de namen en id's van alle trajecten uit de applicatie in alfabetische volgorde?	
Wordt bij klikken op de plus-knop bovenaan doorverwezen naar de pagina om een nieuwe route toe te voegen?	
Wordt er doorverwezen naar de pagina om de gegevens van het traject aan te passen als je op het potloodje naast een traject in de lijst klikt?	
Wordt er doorverwezen naar de pagina om tussenpunten en handlers toe te voegen als er op de naam van een traject in de lijst geklikt wordt?	
Wordt er doorverwezen naar de pagina om tussenpunten en handlers toe te voegen als er op het 2e icoontje naast het traject in de lijst geklikt wordt?	
<b>Trajecten toevoegen</b>	
Wordt bij klikken op de plus-knop bovenaan doorverwezen naar de pagina om een nieuwe route toe te voegen?	
Wordt er doorverwezen naar de pagina om tussenpunten en handlers toe te voegen als er op het 2e icoontje naast het traject in de lijst geklikt wordt?	
Worden de correcte geolocaties aan het begin- en eindpunt toegevoegd als je deze punten op de kaart aanduidt?	
Komt het nieuwe traject met de correcte naam in de lijst als je op 'toevoegen' klikt?	
<b>Trajecten verwijderen</b>	
Wordt het correcte traject effectief uit de lijst verwijderd als er op het kruisje naast het traject geklikt wordt?	
<b>Trajecten aanpassen</b>	
Kan je de naam en thresholds van het traject wijzigen?	
Wordt er verhinderd om de waarde van de threshold van het laagste level (0) aan te passen?	
Wordt er verhinderd dat een hoger level een lagere threshold heeft?	
Worden de wijzigingen opgeslagen als er op 'wijzig' wordt geklikt?	
Worden de wijzigingen genegeerd als er op de pijl om terug te keren geklikt wordt?	
Wordt er doorverwezen naar de pagina om tussenpunten en handlers toe te voegen nadat de wijzigingen gebeurd zijn?	
<b>Tussenpunten en handlers aan trajecten toevoegen?</b>	
Kan er een tussenpunt toegevoegd worden door op 'voeg tussenpunt toe' te klikken?	
Wordt de correcte geolocatie toegevoegd als je het punt op de kaart aanduidt?	
Wordt er verhinderd dat het tussenpunt voor het beginpunt kan ingevoegd worden?	
Wordt er verhinderd dat het tussenpunt na het eindpunt kan ingevoegd worden?	
Wordt het tussenpunt op de correcte plaats toegevoegd in de lijst als er op 'toevoegen' geklikt wordt?	
Wordt een tussenpunt uit de lijst verwijderd als er op het kruisje naast het tussenpunt geklikt wordt?	
Kan er een handler aan een thresholdlevel toegevoegd worden door op 'voeg nieuwe melding toe' te klikken?	
Wordt er verhinderd dat 2 keer dezelfde handler aan hetzelfde niveau wordt toegevoegd?	
Worden alle handlers verwijderd als er op 'reset meldingen' wordt geklikt?	
Wordt er door bovenaan op 'wijzig' te klikken doorverwezen naar de pagina om de gegevens van het traject aan te passen?	



## Systeem

Systeem	
Kunnen de waarden van de systeemvariabelen worden aangepast?	
Is er een validatie op deze waarden?	
Worden de wijzigingen opgeslagen als er onderaan op 'wijzig' wordt geklikt?	
Worden de wijzigingen genegeerd als er bovenaan op de pijl om terug te keren geklikt wordt zonder op 'wijzig' te klikken?	

## Logs server

Systeem	
Wordt er een lijst weergegeven met de laatste 100 logs die de server geregistreerd heeft met de meest recente logs bovenaan?	

## Gebruikers

Gebruikers	
Wordt er een lijst met de naam en de gebruikersnaam van alle gebruikers weergegeven?	
Kunnen de gegevens voor een nieuwe gebruiker worden ingevuld?	
Wordt het wachtwoord verborgen bij het ingeven?	
Wordt de nieuwe gebruiker toegevoegd aan de lijst als er op 'genereer nieuwe gebruiker' geklikt wordt?	

## Profiel

Profiel	
Worden de correcte gegevens van het account waarmee je bent aangemeld weergegeven zonder dat het wachtwoord getoond wordt?	
Kunnen de gegevens aangepast worden en opgeslagen door op 'toevoegen' te klikken?	
Blijft het wachtwoord ongewijzigd als je dit veld leeg laat bij de aanpassingen?	
Wordt het wachtwoord verborgen bij het ingeven?	

## API-Keys

API-Keys	
Wordt er een lijst weergegeven met alle API-Keys die aangeeft of deze actief is of niet?	
Kan een API-Key gedeactiveerd (actief=0) worden door op 'deactiveren' te klikken naast de sleutel in de lijst?	
Wordt er een nieuwe, actieve (actief=1) API-Key aan de lijst toegevoegd als er op 'genereer nieuwe sleutel' geklikt wordt?	

## Hoofdstuk 9: Referentielijst

---

<http://apiary.io/> (Apiary Inc., 2016)

<http://sass-lang.com/install> (Sass, 2006-2015)

<http://materializecss.com/> (Materialize, 2014-2016)

<http://www.ocpssoft.org/docs/prettyfaces/3.3.3/en-US/html/> (PrettyFaces, 2016)

<http://www.jqueryscript.net/time-clock/Pretty-Date-Time-Picker-Plugin-For-Bootstrap-Material.html> (T00rk, 2016)

<http://demo.geekslabs.com/materialize/v2.2/layout02/index.html> (Geekslabs, 2015)

<http://tympanus.net/codrops/2013/08/28/transitions-for-off-canvas-navigations/> (Mary Lou, 2013)

<http://tablesorter.com/docs/#Demo> (Christian Bach, 2016)

<http://blog.stevenlevithan.com/archives/date-time-format> (Steven Levithan, 2007)

<http://www.journaldev.com/7252/jsf-authentication-login-logout-database-example> (Pankaj, 2015)

<https://github.com/liskiew/jquery-geolocation-picker> (Grzegorz Blaszczyk, 2011)

<http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api> (Vinay Sahni, 2013)

<http://www.oracle.com/technetwork/java/javaee/overview/index.html> (Java EE, 2016)

<http://leafletjs.com/reference.html> (Leaflet, 2016)

<https://www.mapbox.com/api-documentation/> (Mapbox, 2016)

<https://developers.google.com/maps/web-services/> (Google, 2016)