

## Sollicitatieopdracht

De Dienst Infrastructuur Verkeer en Vervoer (DIVV) van de Gemeente Amsterdam meet de actuele reistijden op de autowegen die zij beheert. Deze data is beschikbaar op de volgende URL:

<http://www.trafficlink-online.nl/trafficlinkdata/wegdata/TrajectSensorsNH.GeoJSON>

Het geretourneerde JSON bestand bevat per route de volgende velden:

- **properties**
  - **LOCATION:** De naam van het traject
  - **TIMESTAMP:** Timestamp voor het moment waarop reistijd bepaalt is
  - **COLOR:** De kleur die in het verkeerssysteem wordt gebruikt voor de huidige reistijd
  - **VELOCITY:** Huidige snelheid (km/u) op traject
  - **LENGTH:** Lengte van traject (in meters)
  - **TRAVELTIME:** De huidige reistijd op het traject
  - **TRAVELTIME\_FF:** 'Traveltime Free Flow', de reistijd bij normale doorstroming
- **geometry**
  - **coordinates:** De coördinaten van het traject, in het [Rijksdriehoekskoördinaten](#) stelsel en in de volgorde van de rijrichting

## Opdracht

1. Ontwikkel een applicatie gebaseerd op Java, die de actuele reistijden ophaalt en opslaat in een database. Zorg, op de manier die voor jou het handigst is, dat de gegevens om de 5 minuten worden ververs in de database.
2. Schrijf een REST service, op de manier die voor jou het handigst is, die op basis van een bepaald traject (LOCATION) de waardes van het afgelopen uur voor de snelheid (VELOCITY) en reistijd (TRAVEL\_TIME) uit de database teruggeeft.
3. Schrijf een Javascript applicatie die, gebruik makend van de eerder geschreven REST service, de snelheid en reistijd van alle trajecten in de tijd weergeeft. Je kunt kiezen uit een van de volgende visualisaties, van makkelijk naar moeilijk:
  - a. Toon een tabel waarin per traject de snelheid en reistijd van het afgelopen uur per 5 minuten wordt weergegeven, aangevuld met de overeenkomstige kleur.
  - b. Toon een grafiek waarin het verloop van de snelheid en reistijd per traject in het afgelopen uur wordt gevisualiseerd. Kies een grafiek die deze data het beste weergeeft.
  - c. Toon de reistijd en snelheid in een kaart en geef trajecten de juiste kleur die overeenkomt met de huidige reistijd. Voeg de mogelijkheid toe om stapjes terug in de tijd te gaan, op dezelfde manier als bijvoorbeeld [Buienradar](#).
  - d. Maak een combinatie van bovenstaande visualisaties of bedenk een originele representatie.

## **Uitwerking**

1. Begin met het maken van een inschatting van de benodigde tijd om een werkende (web)applicatie voor bovengenoemde opdracht te realiseren en koppel deze terug naar Finalist.. Op basis daarvan spreken we gezamenlijk een datum voor de presentatie af.
2. Werk de opdrachten uit tot een werkende (web)applicatie. Houd bij het maken van de opdracht een logboekje bij van de stappen die je gezet hebt en de tijd die je per stap besteed hebt. Lever je broncode een aantal dagen voor je presentatie op in GitHub (met instructies om het werken te krijgen).
3. In een presentatie van maximaal 45 minuten licht je jouw oplossing toe voor de klant. Maak duidelijk hoe je te werk bent gegaan en onderbouw de verschillende keuzes die je hebt gemaakt. Als je zelf geen laptop meeneemt/hebt, geef dat vooraf even aan dan kan daar enigszins rekening mee worden gehouden.

Voor de uitwerking van de opdrachten heb je de vrijheid om zelf te kiezen voor:

- Inzet van een Java framework. Indien je een framework in wilt zetten, ben je vrij in de keuze daarin (Spring, GWT, ...).
- Inzet van een Javascript framework en/of aanvullende Javascriptlibraries. Indien je een framework in wilt zetten, ben je vrij in de keuze daarin (jQuery, AngularJS, ...).
- Inzet van een specifieke database (bijvoorbeeld MySQL, MongoDB, CouchDB, ...), die naar jouw mening geschikt is voor dit probleem en de meeste toegevoegde waarde oplevert.
- Extra toevoegingen aan de originele opdracht, waarvan jij denkt dat er een betere oplossing uit komt voor de klant.
- Licht je keuzes toe.