

Distributed Systems Assignment 1: Web Services

Tom De Schepper, Glenn Daneels, and Steven Latré

2018-2019

1 Introduction

This document discusses the first assignment for the practical part of the Distributed Systems course. The goal of this assignment is to learn how to successfully utilize existing web services and create one of your own. The assignment will need to be completed individually. In this document we will provide you with a detailed list of the features and requirements that are needed.

2 Goal

The goal of this assignment is twofold:

1. First, you will create your own (RESTful) web service in Python that will consume existing web services. In particular you will use the data API of the Flemish public transport company to get the planned schedule of different bus and trams and calculate the real-time positions of the different vehicles. Furthermore, you will use weather web service to illustrate the weather on each stop.
2. Second, you will create a web front-end that will consume the provided web service and show the moving vehicles, the stops and the weather at those stops for a certain line on a map. Basically, something similar to what already exists for trains in Belgium (<https://trainmap.belgiantrain.be/>).

3 Requirements

In this section we will list the different features and components that will need to be implemented in the assignment.

3.1 Web service

The purpose of the RESTful web service is to return (using the json format), for a given tram or bus line, a list with the (geographical) positions of all vehicles on that particular line at the current moment. Furthermore, a list with all the stops of the line with the needed info (e.g., name, coordinates, ...) needs to be returned, as well as an indication of the weather at the stops. In order to create this web service and its functionality you will need to use a number of existing web services or APIs:

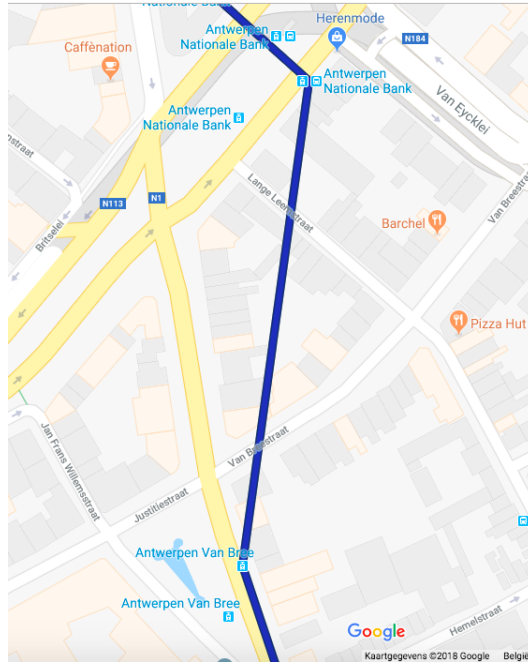


Figure 1: Google maps connection of two consecutive stations

- The data API from DeLijn will need to be used to get the planned schedule for the line and its stops (<https://data.delijn.be>).
- A web service offering weather information for specific locations.
- A directions API to help estimate the route between two stops (more on this in the next paragraph).

It is your job to combine the data offered from the different web services / data APIs and create some additional value (e.g., the position of the different vehicles and the weather at the different stops). In your manual, you must clearly state what your web service offers and how it can be accessed.

As the API of the DeLijn gives you only the upfront scheduled time a bus or tram should be at a certain stop you will have to implement some logic that estimates the real position of the buses and trams at a specific time. The straightforward method is to assume a direct line between two consecutive stations and then estimate the position on the time schedule. This assumption of a direct line between two stations, is actually how Google Maps works currently, as seen in Figure 1. However, this can lead to positioning a bus or tram in the middle of a building. Therefore, you will have to come up with a more intelligent solution that can be used to estimate the positions as realistically and correct as possible. For this you might want to use a directions API that can give you detailed routes between two places.

Finally, some notes:

- You need to take into account that the line numbers of the DeLijn are not unique across Flanders (e.g., there is a tram line with number 2 in both Antwerp and Ghent).

- In order to bring your web service online, you will use the Python Flask library.
- You are allowed to extend the functionality offered by your web service if you want to (but be sure to focus on the requirements first).

3.2 Front-end

The second part of the assignment is to make a simple website that actually consumes the created web service. On this web page you need to be able to specify a certain line and show in real-time a map with the current vehicles on this line. Furthermore, all stations need to be marked on the map and for each station a weather indication is present.

For this small website you can make use of technologies/languages such as HTML and Javascript. To have a nice layout without a lot of effort, you can use, for instance, the Bootstrap library. For the map itself you can make use of an existing map API (e.g., Google Maps). More information about the different tools can be found in the next section.

Finally, it is not necessary to host this website on a web server. For this assignment, it is sufficient that your website can be opened locally in your browser.

4 Tools

As already hinted on in the previous section you will use a number of tools, technologies, and languages to successfully solve this assignment. While some of these are mandatory, others can be used based on your own preferences. For the existing web services like the data API of DeLijn, the map/direction service, and the weather service you will need to create an account and acquire an API key. Below you will find a list of tools and some information.

- Python and Flask: the web service will need to be created in Python using the Flask microframework (<http://flask.pocoo.org/>), which is very straightforward to setup. When using the PyCharm IDE, Flask should normally already be available.
- DeLijn API: this needs to be used to acquire the needed public transport information. An account can be created at the following website: <https://data.delijn.be/signup>
- Map/direction service: needed for the estimation of the position of vehicles and to show a real-time map. You are free to choose the service you will utilize. The most known and extensive service is the Google Maps Platform (<https://developers.google.com/places/web-service/intro>). However, in order to use this service you need to provide a credit card number, even when using free of charge services like the Map and Direction APIs. An alternative is Open Street Map (https://wiki.openstreetmap.org/wiki/Main_Page).
- Weather service: once again you are free to choose which existing weather service you will use. One possibility is, among others, Open Weather Map (<https://openweathermap.org/>).
- Web technologies and languages: for the construction of your website you are also free to choose what technologies, libraries, or languages you want to use. The typical ones you might want to use are HTML (for the structure of the website), CSS (for the layout of the website), and Javascript (for adding dynamic functionalities to the

website). As mentioned, it can be interesting to look at the Bootstrap library (<https://getbootstrap.com/>) to fast forward your website design.

For all of the listed tools, technologies, and libraries, there is an extensive amount of documentation online waiting for you to be consulted.

5 Minimal requirements

There are some minimum requirements that your solution should comply to, in order to pass for this part of the course. If this requirements are not fulfilled, it is not possible to pass:

- A manual needs to be present in your submission or we will not be able to evaluate your solution. This manual needs to describe (minimally, but correctly) the possible web service calls, how we can run and test your solution and an extensive explanation of the design choices you made and which tools were used.
- In your manual there needs to be a complete definition of your web service(s) (including name, parameters, and output structure). For the definition of the output structure, you can, for instance, look at how this is done by the data API of the DeLijn .
- It needs to be able to run your solution (e.g., start up your web service, open the website, ...). Make sure that your solution is complete upon submission and no files or libraries are missing.
- Make sure to only use web services and web communication (e.g., HTTP calls) within this assignment and no other communication methods.

Remark: if your solution complies to these minimum requirements, this does not mean that you will certainly pass this assignment.

6 Deadline and submission

The deadline for this assignment is Wednesday 21st of November at 23:59. Your solution has to be submitted through blackboard.