

Manual

Beau De Clercq

July 2019

Contents

1	Running & using the app	2
1.1	Starting the app	2
1.2	Usage	2
2	API calls	3
2.1	Provinces	3
2.2	Lines	4
2.3	Routes & passings	5
2.4	Other calls	5
3	Site links	6
3.1	/	6
3.2	/provinces	7
3.3	/lines/<province>	8
3.4	/timetable/<province>/<line>/to	8
3.5	/timetable/<province>/<line>/from	9
4	Design choices	10
5	Tools used	11

1 Running & using the app

1.1 Starting the app

In order to run the app, the user needs to execute the *run.sh* script on Mac/Linux or the *run.bat* script on Windows.

1.2 Usage

2 API calls

2.1 Provinces

```
{"data":{  
    "provinces":[  
        {"entiteitcode":"A",  
        "entiteitnummer":1,  
        "links": [  
            {"rel":"detail","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/1"},  
            {"rel":"haltes","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/1/haltes"},  
            {"rel":"lijnen","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/1/lijnen"},  
            {"rel":"gemeenten","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/1/gemeenten"}],  
        "omschrijving":"Antwerpen"},  
        {"entiteitcode":0,  
        "entiteitnummer":2,  
        "links": [  
            {"rel":"detail","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/2"},  
            {"rel":"haltes","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/2/haltes"},  
            {"rel":"lijnen","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/2/lijnen"},  
            {"rel":"gemeenten","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/2/gemeenten"}],  
        "omschrijving":"Oost-Vlaanderen"},  
        {"entiteitcode":B,  
        "entiteitnummer":3,  
        "links": [  
            {"rel":"detail","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/3"},  
            {"rel":"haltes","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/3/haltes"},  
            {"rel":"lijnen","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/3/lijnen"},  
            {"rel":"gemeenten","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/3/gemeenten"}],  
        "omschrijving":"Vlaams-Brabant"},  
        {"entiteitcode":L,  
        "entiteitnummer":4,  
        "links": [  
            {"rel":"detail","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/4"},  
            {"rel":"haltes","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/4/haltes"},  
            {"rel":"lijnen","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/4/lijnen"},  
            {"rel":"gemeenten","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/4/gemeenten"}],  
        "omschrijving":"Limburg"},  
        {"entiteitcode":W,  
        "entiteitnummer":5,  
        "links": [  
            {"rel":"detail","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/5"},  
            {"rel":"haltes","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/5/haltes"},  
            {"rel":"lijnen","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/5/lijnen"},  
            {"rel":"gemeenten","url":"https://api.delijn.be/DLKernOpenData/api/v1/entiteiten/5/gemeenten"}],  
        "omschrijving":"West-Vlaanderen"}]  
,  
  }  

```

API call:

/get_provinces

Return value: a JSON format response object as shown above.

2.2 Lines

```
{"data":{  
    "lines": [  
        {"bedieningtype": "NORMAAL", "entiteitnummer": "1",  
        "lijnGeldigTot": "2019-08-31", "lijnGeldigVan": "2019-04-22",  
        "lijnnummer": "2", "lijnnummerPubliek": "2",  
        "links": [{"rel": "detail", "url": "https://api.delijn.be/DLKernOpenData/v1/beta/lijnen/1/2"}],  
        "omschrijving": "Hoboken - P+R Merksem", "publiek": true, "vervoertype": "TRAM"},  
        {"bedieningtype": "NORMAAL", "entiteitnummer": "1",  
        "lijnGeldigTot": "2019-08-31", "lijnGeldigVan": "2019-04-22",  
        "lijnnummer": "3", "lijnnummerPubliek": "3",  
        "links": [{"rel": "detail", "url": "https://api.delijn.be/DLKernOpenData/v1/beta/lijnen/1/3"}],  
        "omschrijving": "P+R Merksem - P+R Melsele", "publiek": true, "vervoertype": "TRAM"},  
        {"..."},  
        {"bedieningtype": "NORMAAL", "entiteitnummer": "1",  
        "lijnGeldigTot": "2019-08-29", "lijnGeldigVan": "2019-06-02",  
        "lijnnummer": "997", "lijnnummerPubliek": "Buu",  
        "links": [{"rel": "detail", "url": "https://api.delijn.be/DLKernOpenData/v1/beta/lijnen/1/997"}],  
        "omschrijving": "Buurtbus Brasschaat", "publiek": true, "vervoertype": "BUS"},  
        {"bedieningtype": "TECHNISCHE_LIJN", "entiteitnummer": "1",  
        "lijnGeldigTot": "2019-08-31", "lijnGeldigVan": "2019-04-22",  
        "lijnnummer": "999", "lijnnummerPubliek": "999",  
        "links": [{"rel": "detail", "url": "https://api.delijn.be/DLKernOpenData/v1/beta/lijnen/1/999"}],  
        "omschrijving": "Uit- en inrijdende ritten Antwerpen Tram", "publiek": false, "vervoertype": "BUS"}  
    ],  
    "status": "success"  
}  
}
```

API call:

```
/get_lines/<province>
```

Return value: a JSON format response object as (shortened) shown above.
This object describes the available lines for the given province.

2.3 Routes & passings

```
{"data":{  
    "routes": [  
        ["51.2587202","4.4627988"], ["51.2580442","4.4624877"], ["51.2560165","4.4605994"],  
        . . . ,  
        ["51.1828995","4.3736422"], ["51.1828244","4.373138"], ["51.1827684","4.3728861"]  
    ],  
    "stops": [  
        {"lati":51.258726674685,"longi":4.46271548234936,  
        "name":"P+R Merksem",  
        "temp":17.45,"time":["00:35:00","06:35:00"],"weather":"clear sky"},  
        {"lati":51.25366088473669,"longi":4.458164614670439,  
        "name":"Ringlaan",  
        "temp":17.45,"time":["00:36:00","06:36:00"],"weather":"clear sky"},  
        {"lati":51.251262697852084,"longi":4.45592564903742,  
        "name":"Rerum Novarum",  
        "temp":17.45,"time":["00:37:00","06:37:00"],"weather":"clear sky"},  
        { . . . },  
        {"lati":51.18318020637612,"longi":4.383688211677621,  
        "name":"Schijfwerper",  
        "temp":17.44,"time":["01:06:00","07:06:00"],"weather":"clear sky"},  
        {"lati":51.18309990244975,"longi":4.378267845704564,  
        "name":"Sportstraat",  
        "temp":17.44,"time":["01:08:00","07:08:00"],"weather":"clear sky"},  
        {"lati":51.18317216965261,"longi":4.37246136797233,  
        "name":"Zwaantjes",  
        "temp":17.42,"time":["01:09:00","07:09:00"],"weather":"clear sky"}  
    ],  
    "status":"success"  
}
```

API calls:

```
/get_timetable/<province>/<line>/to/  
/get_timetable/<province>/<line>/from/
```

Return value: a JSON format response object as (shortened) shown above. This object describes the stops on the line and provides passing times, weather information and coordinates for the route that will be followed.

2.4 Other calls

API calls:

```
/timetable/static/img/<name>  
/timetable/<province>/<line>/static/img/<name>
```

Return value: the image saved in the folder *static/img* as *name*.

3 Site links

3.1 /

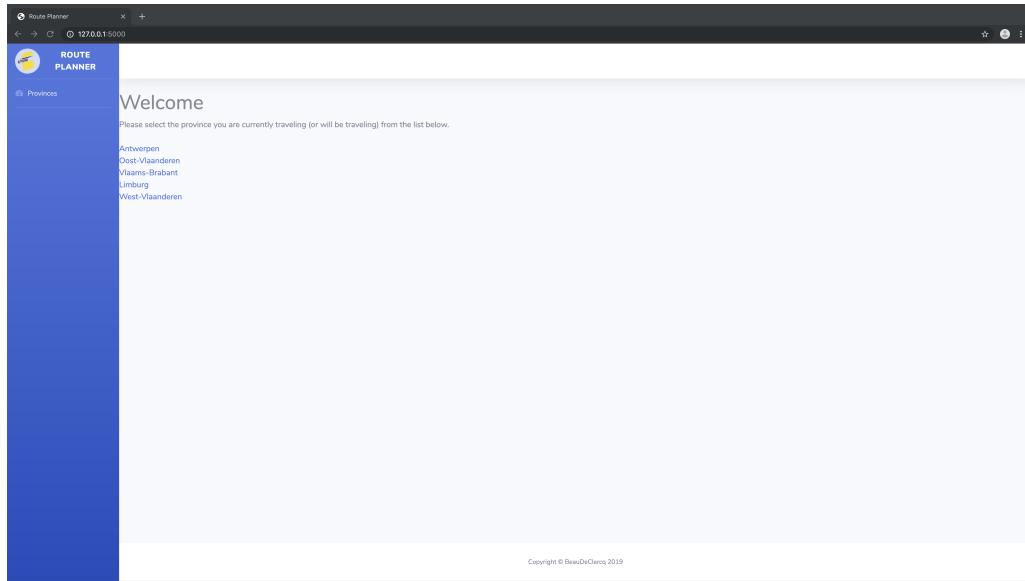


Figure 1: Home page

At the home page, a user can start the process of planning his/her route by selecting a province.

3.2 /provinces

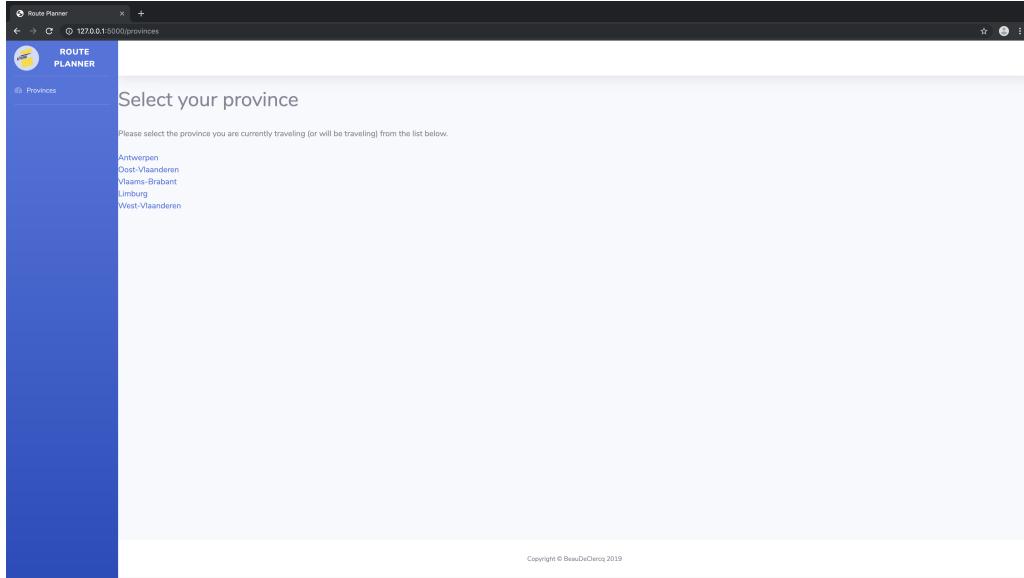


Figure 2: Province page

At the ‘Provinces page’, a user can start the process of planning his/her route by selecting a province. This will take the user to the next step.

3.3 /lines/<province>

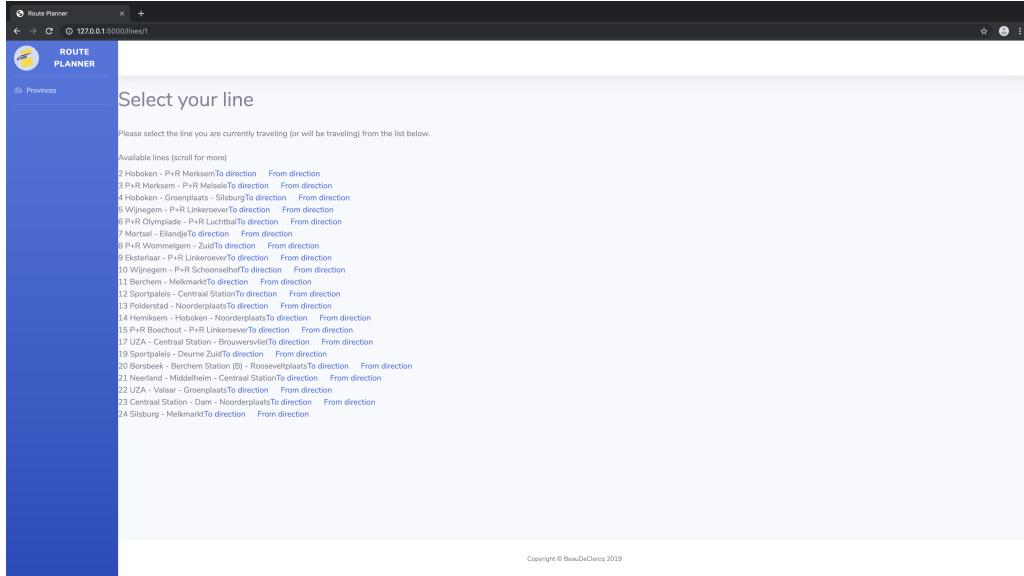


Figure 3: Line selector

After having selected a province, the user is provided a list of all lines that are available for that province.

On the 'Lines page', the user can then select which direction he or she wants to go.

3.4 /timetable/<province>/<line>/to

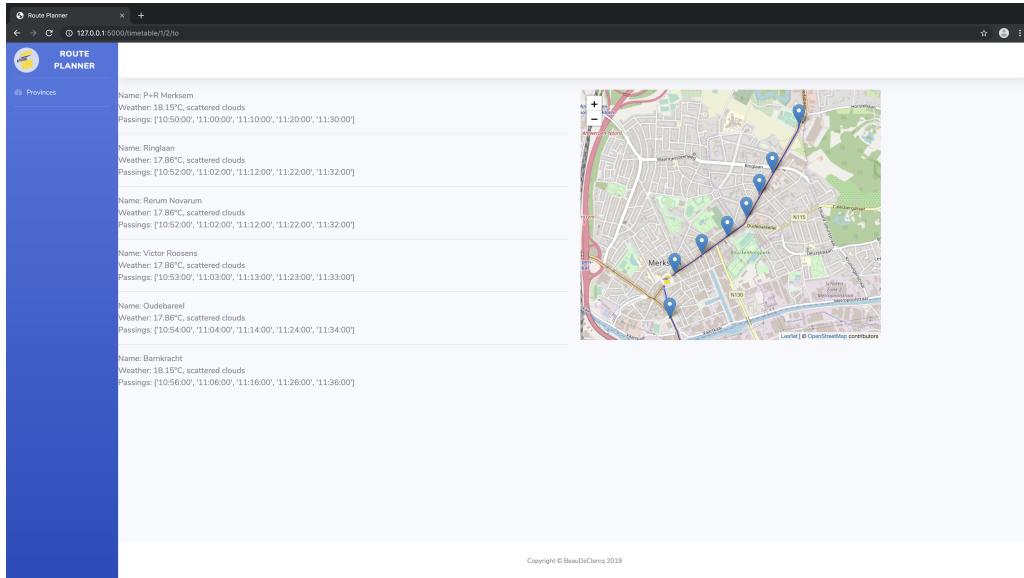


Figure 4: Example route

Once a direction has been chosen, everything will be processed and the user will be given a page containing all necessary info.

3.5 /timetable/<province>/<line>/from

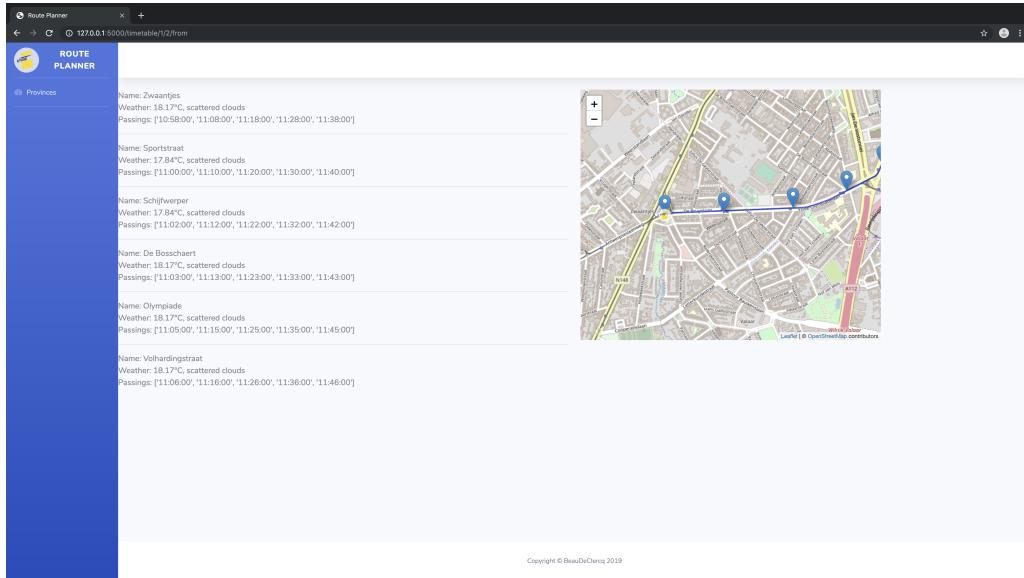


Figure 5: Example route

Once a direction has been chosen, everything will be processed and the user will be given a page containing all necessary info.

4 Design choices

5 Tools used

- Flask as the main framework
- Python *requests* module to reduce the lines of code needed to perform an API call
- JavaScript & LeafLet.js for all things map-related