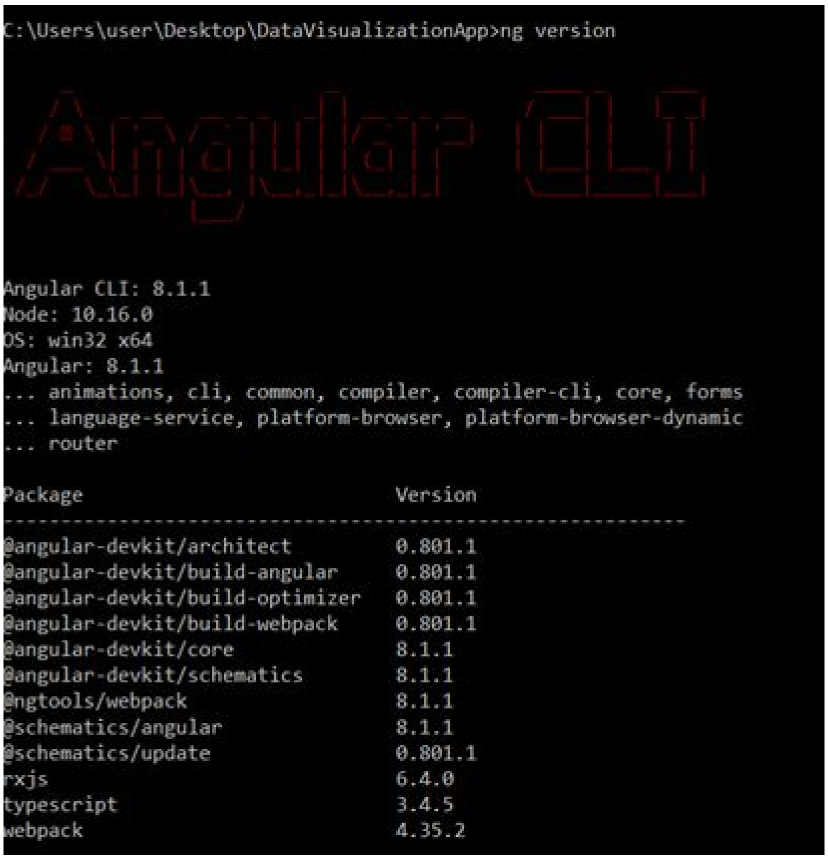
# DATA VISUALISATION
# ANGULAR APPLICATION

## <u>Prerequesite :</u>

Make sure your Angular and Angular CLI are both at least v.8.1.1 before doing anything.

("ng version" command to check version)

NodeJS shouldn't be too far from 10.16 as well.



```
C:\Users\user\Desktop\DataVisualizationApp>ng version

Angular CLI: 8.1.1
Node: 10.16.0
OS: win32 x64
Angular: 8.1.1
... animations, cli, common, compiler, compiler-cli, core, forms
... language-service, platform-browser, platform-browser-dynamic
... router

Package                           Version
-----------------------------------------------------------
@angular-devkit/architect         0.801.1
@angular-devkit/build-angular     0.801.1
@angular-devkit/build-optimizer   0.801.1
@angular-devkit/build-webpack     0.801.1
@angular-devkit/core              8.1.1
@angular-devkit/schematics        8.1.1
@ngtools/webpack                  8.1.1
@schematics/angular               8.1.1
@schematics/update                0.801.1
rxjs                              6.4.0
typescript                        3.4.5
webpack                           4.35.2
```

## Running the project :

To run the project, the only thing you should have to do is to clone it from : https://github.com/Zakouille/data-visual-app

And

Run "npm install" in the app’s directory.

Then just run "ng-serve" and you should be good to go at http://localhost:4200/ by default.

If any problem occurs with “chart.js”, please refer to the installation commands bellow.

## Understanding the application :

The application is composed of 3 components and a service.

Let’s list them and slightly talk about each of these :

App.component : There in all Angular apps. Doesn’t have much used in this project besides « assembling » the HTML code of other components.

```
<div class=wrapper>
  <app-header></app-header>
  <div class="main-panel">
    <div class="content">
      <app-dashboard></app-dashboard>
    </div>
  </div>
</div>
```

Header.component : contains Siemens’s logo for now but can contain anything that should be at the top of the page.

Dashboard.Component : contains everything else that is printed for now. It contains the printing of the dashboard on the HTML side as well as the data getting process on the typescript side. It uses the « getting-data » service to receive the data he needs to display those.

Getting-data.service : contains the instance of the Http Client used to send request to the backend. Everything related to requesting data should go in there.

Finally, a last file we should take a look at is « environment.ts » where are setup any URLs used in our application.

## Back-end

### API URL

https://github.com/musmastour/data-vizualizer.git

### Routes

→ /generate/data

This route will generate data to print into the dashboard. There is the data that will be generated:

- temperatureOutside
- lights
- fryingTemperature
- gasConsumption
- 
- powerConsumption :
    - motor1
    - motor2
    - motor3

→ /get/output/potatoes

This route will generate 31 random values between 100 and 200 and represente output potatoes a day. Those data will be printed on the chart

## FRONT-END

### Chart.js

### Installation

```
npm install --save chart.js
```

### Install the zoom feature

```
npm install --save chartjs-plugin-zoom
```

add those files to index.html

```
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>
<script src="https://cdn.jsdelivr.net/npm/hammerjs@2.0.8"></script>
<script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-zoom@0.7.3"></script>
```

### How it works

Init your chart like this :

```
this.lineChart = new Chart(speedCanvas, config);
```

here speedCanvas is an HTML element

and precise your configurations :

```
var config = {
      type: 'line',
      data: {
```

```
  labels: month, // Date Objects
  datasets: [{
    label: 'Potatoes',
    data: this.dataChart,
    fill: false,
   borderColor: '#081e26',
   backgroundColor: 'transparent',
   pointBorderColor: '#081e26',
   pointRadius: 4,
   pointHoverRadius: 4,
   pointBorderWidth: 8,
 }, ]
},
options: {
  responsive: true,
  scales: {
    xAxes: [{
      type: 'time',
      time: {
        parser: this.timeFormat,
        tooltipFormat: 'll HH:mm'
      },
      scaleLabel: {
        display: true,
        labelString: 'Date'
      },
      ticks: {
        maxRotation: 0
      }
    }],
    yAxes: [{
      scaleLabel: {
        display: true,
        labelString: 'Value'
      }
    }]
  },
  plugins: {
   zoom: {
     pan: {
       enabled: true,
       mode: 'x'
```

```
          },
          zoom: {
            enabled: true,
            mode: 'x',
          },
        }
      }
    }
  };
```

Here, **labels** will be data printed on the x axe.
On the **dataset**, you will have to but data which will correspond to labels.
On the **option** section, write information about the scale, the zoom options axes type and other stuff

more information here : https://www.chartjs.org/docs/latest/charts/line.html
and for the zoom: https://www.npmjs.com/package/chartjs-plugin-zoom

Mustapha Mastour
Zakary Nizard