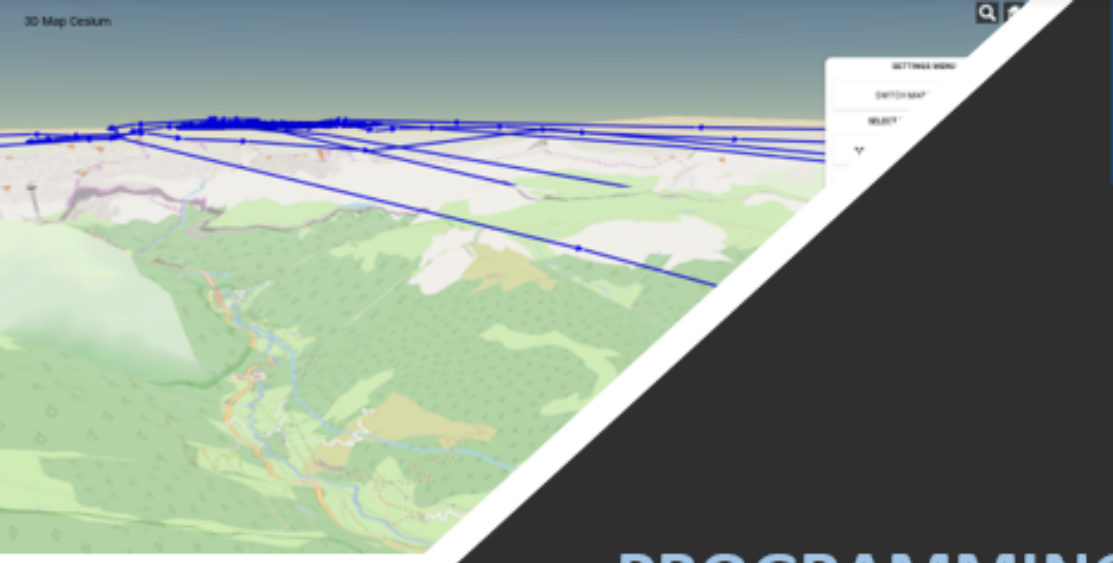
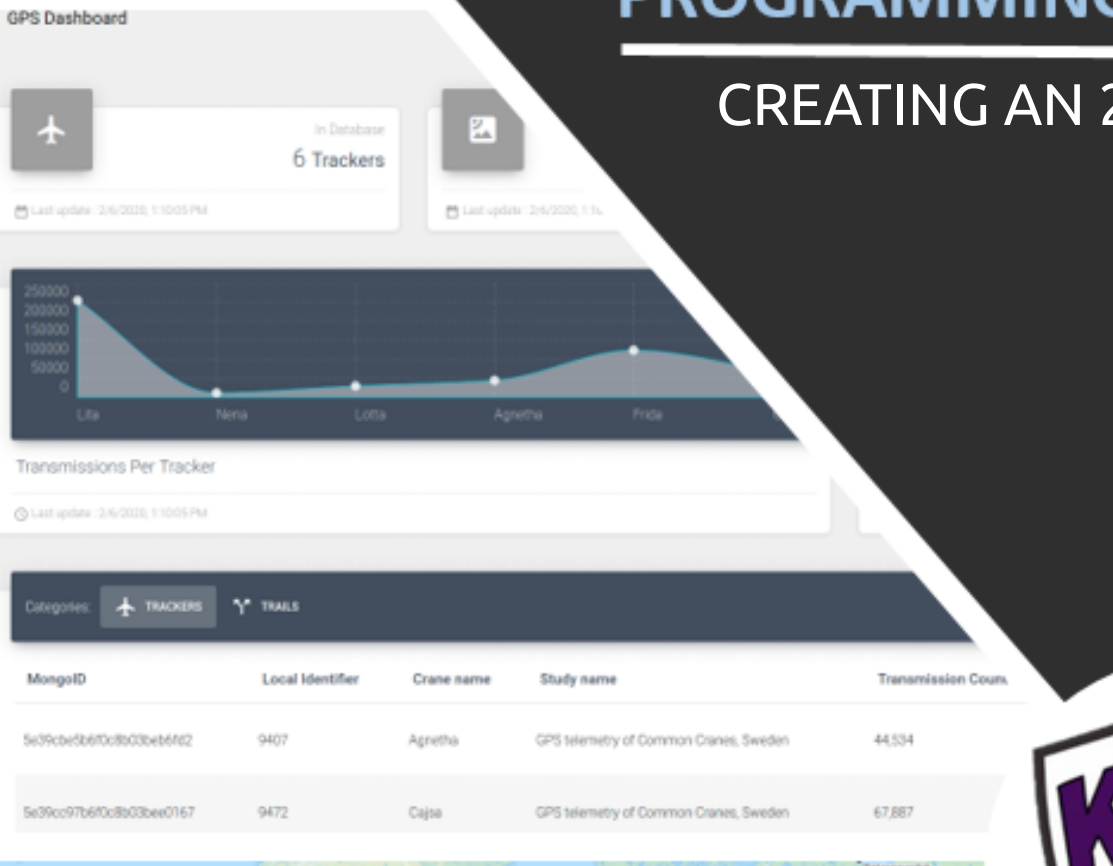


Ministerie van Defensie

PROGRAMMING MANUAL

CREATING AN 2 DIMENSIONAL MAP VIEWER



overtref jezelf

Date: 10-02-2020
Version: 1.0
Author: Pieter Lems
Department: DMO/JIVC/KIXS

Purpose of this document

This programming manual serves as an extension for the following documents:

1) Cookbook: Creating the Geostack infrastructure

   The datastores, tools and libraries used during this programming manual are installed and created in the cookbook: Creating the Geostack infrastructure.

2) Cookbook: Creating the Angular base application

   The base application of this Dataset Dashboard has been created during the cookbook: Creating the Angular base application.

3) Cookbook: Data modeling in MongoDB using MongoEngine

   The data used during this cookbook, is modeled, indexed and imported in the cookbook: Data modeling in MongoDB using MongoEngine

4) Programming manual: Creating the Flask API/Middleware

   The Middleware that will be used during this programming manual is created in the programming manual: Creating the Flask API/Middleware

If you have not read these documents yet, please do so before reading this document.

The purpose of this programming manual is to create an 2D map viewer application using the Angular Javascript framework and OpenLayers 6. This application is an extension of our Angular base application.

The Angular apps will perform API calls to our Flask application and our Flask application will then retrieve the requested data via queries, performed on our datastores. The results are then returned to our Angular applications.

Table of Contents

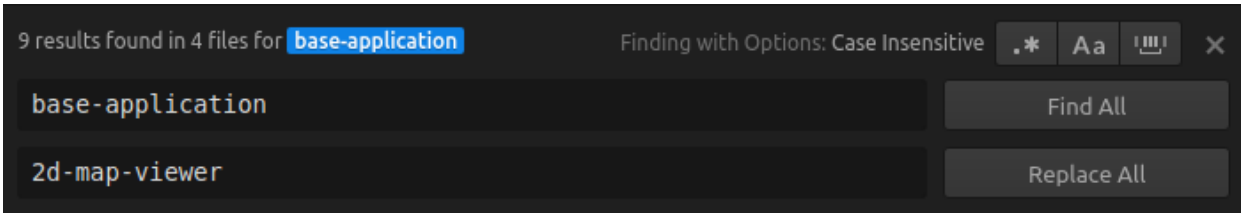
1.Introduction

1.1 Getting ready

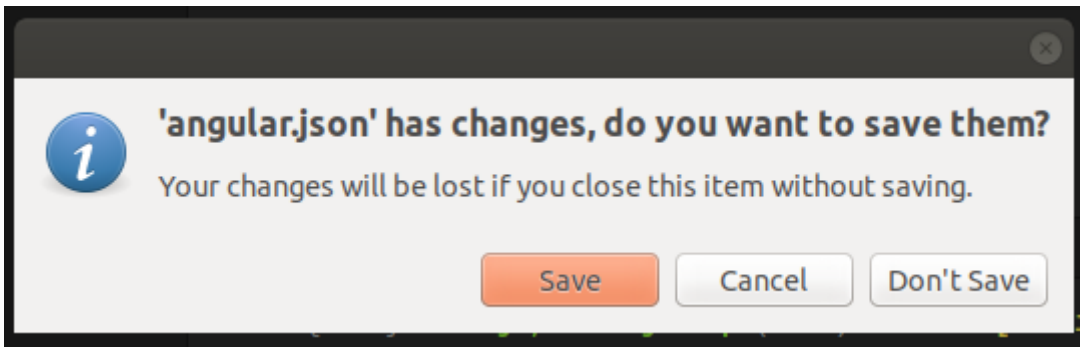
This application is an extension of the base-application so we can copy this application and start creating the 2D map viewer from there. After we copied the base-application folder, we need to change some names and titles to make the new application the 2D map viewer. We start by changing the name of the folder we just copied from base-application to 2d-map-viewer, as shown in the image below.

We also need to edit the project name: “base-application” to “2d-map-viewer”. If you are using the code-editor Atom, this is done by performing the following steps:

1) In the edit press the keys ctrl + shift + f
2) In the screen that pops up enter: “base-application” in the find section and “2d-map-viewer” in the replace section, as shown in the illustration below. Then click on find all.



3) Click on replace all and on the save button in the screen that pops up.



4) In the file: index.html located in the folder 2d-map-viewer/src, replace the title from BaseApplication to 2D Map Viewer.
5) In the file: sidebar.component.html located in the folder src/app/components/sidebar/, change the text: “Base Application” to “2D Map Viewer.”

1.2 Adding the geospatial framework OpenLayers

Copy the spatial framework to assets/geospatial-frameworks/

Add the following lines inside the <head> element from the index.html page located in the folder 2d-map-viewer/src/app/.

```
<!--Here we add the reference to the OpenLayers style sheet-->
<link rel="stylesheet" href="/assets/geospatial-frameworks/OpenLayers/ol.css"/>

<!--Here we add the reference to the OpenLayers javascript code-->
<script src="/assets/geospatial-frameworks/OpenLayers/ol.js"></script>
```

2. Creating the services

2.1 The map service

Create a new file called map.service.ts

2.2 The Crane service

make new file

4. Creating the 2D Map page

Create a new folder in the pages folder, called map-page

In that folder create 2 files:

map.component.ts

map.component.hmtl

4.1 Creating the HTML file

Creating the map container of our OpenLayers Map

add the following to that file:

```
<!--
Here we create a div element to which the OpenLayers map will be assigned
We give the div a height of 100vh (Full screen height) and a width of 100%
which is the full width of the screen. -->
<div id="map" style="height: 100vh; width: 100%;"></div>
```

4.1.1 Creating the base of the settings menu

Adding the base of the settings menu

4.2 Creating the Map Component

Create a new file in the map-page folder called map.component.ts

import modules

declare openlayers contant

create component metadata

create component class

create empty ngoninit

create the timeconverter function

Add the map component to app.module.ts

Now we need to add the GPSDashboard to our app.module.ts, so let's open this file and add an import to the top of the file. The code for this is shown in the illustration below.

Now we need to add a new a new route to our app-routing.module.ts file. So let's open this file. First we need to import the GPSDashboard in this file. This is done by adding the following code to the top of that file.

Then we need to add a new route to our routelist. The code for this is shown in the illustration below.

If you want you can change the route which redirects us to the base-page, to redirect us to the gpsdashboard. If you choose to do this the final routelist will look the same as in the illustration below.

Now we want to add a new entry to our sidebar. This is done in the file: sidebar.component.ts, located in the folder : src/app/components/sidebar/. So let's open this file and add a new route to our routelist. The final code will be the same as shown in the illustration below.

4.2.1 Creating the OpenLayers Map

create global map variable

create global maplayer variable

create global mapProviders jsmap

create getMapProviders function

create createOpenLayersMap() function

add Createmap function to ngonintt

Now when we open the application we have the base map

4.2.2 Switching between map providers.

Add the code setMapProvider

Add the code to the HTML file in a new div element called WMSSelection

4.2.3 Adding items

Create additem function

4.2.4 Retrieving Tracker data

Create global variable items

create getItems function

add the getItems function to the NgOninit

create itemSelection in HTML

now we have a populated list of items in our database

4.2.5 Selecting items

Create global variable selectedItems

create global variable activeItem

create function selecItem()

create function GetInitialItemData()

add check if length of selecteditem is bigger than 0 otherwise the extra content will not be displayed

Add selectedItems to HTML

4.2.6 Loading Item data

Set item data

add HMTL of itemInfoSelection

add function for zooming to location

4.2.7 Creating and setting Layer groups

Create global variable layerStyles

Create Line layer

Create Point layer

Create Marker Layer

Create function setLayerGroups

add function addLayerGroup to function LoadItemData

Add HTML for DTG selection

4.2.8 Creating and setting Overlays

Adding overlay html divs to HTML file

Create function addOverlays()

add Adoverlays function in the CreateMap function()

add setDynamicOverLayes function()

add setStatiocOverlays function()

add setStaticOverlays function to selectItem function and to loaditemdata function and to setLayergroup function

4.2.9 Removing a selected Item

Add fcuntion removeItem

4.2.9 Removing a LayerGroup

Add function removeLayerGrooup

4.2.9 Adding DTG selection

Create function for getting DTG item data

Create DTG picker component folder

create HTML DTG Picker

Create TS DTG picker

Import and add datepicker to app.module.ts

add dateRange as global variable

add getDTGEvent function

add html to dtgSelection

add this.dateRange = this.activeItem.dateRangeTotal; to the selectItem function

4.2.10 Adding Amount selection

Create function for getting selection amount

Add HTML for amount selection

Create global variable items

4.2.11 Adding Country selection

Add global variable country selection

Create function for getting selection by country

Add HTML for amount selection

4.2.12 Adding Layer and Overlay Toggling

Add toggleLayer function

add toggleOverlay function

add HTML

4.2.13 Changing layer styling

Add global colorlist, widthlist , linestylelist

Add global dictionary styleDict

Add function setLayerStyles()

Add HTML

4.2.14 Animating routes

Add animateRoute function

Add clea Animartion fucntion

add HMTL

4.2.15 Creating an elevation profile

Add globals private elevationProfile:any;

 private elevationProfileOpen:boolean;

Add loadElevationData function

Add createLEevationProfile fucntion

add html