Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

geoinformation engineering ikg eth zurich

David Rudi, Christian Sailer

GIS III Assignment 2

Create an interactive Web App of your SenseBox data
about a topic of your choice

1 Introduction

Goal

In this assignment, you should apply the knowledge you gained in the Geosensors exercises in order to develop a full-fledged interactive web client with the SenseBox data, about some topic of your choice. The data must include those collected in class and may include those available online.

Workplaces

You can work in any of the three IKG StudiLabs (HIL G21, HIL G10.5) or at home.

Absences and Repetition

Justified and proved absences (e.g., due to illness) of at least one-week lead to deadline extension of the same duration.

Feedback meeting

You will receive comments on your solution in Moodle. In case you would like to discuss your solution with one of the assistants, please write us an e-mail.

2 Submission

Deadline: **20.12.2018, 23:55**

In Moodle: https://moodle-app2.let.ethz.ch/mod/assign/view.php?id=273846

Submission documents:

- Source code: pack your project folder to a ZIP file (via the file system).
- We expect your source code to be documented in such a way, that there is no need for an additional descriptive document (see also: Grading – Documentation).

3 Task

Develop a responsive Web App Application (e.g., using the ArcGIS Javascript or OpenLayers library) which monitors changes in the environment. Feel free to use any online code editor and open-source learning environment of your choice for development (e.g., https://codepen.io, https://codesandbox.io/, http://collabedit.com).

The Web App has to integrate existing map services about your home area (e.g. City of Zurich or its surrounding areas) from external open data sources, visualize cartographic content in a skillful manner (for which each layer should be edited as far as possible to fit your purpose) and displays (a subset) of the SenseBox data which was collected (by your or your fellow students) in the labs. Information displayed should be targeted at a topic of your choice. Optionally, you can utilize data beyond our own (e.g., via OpenSenseMap).

Describe the purpose/goal of the Web App and the reason for choosing the specified geographical area with its containing layers in your app. Add a section with a "Legal notice" (see the legal section in the Business aspects lecture), a section with "About us" where you describe yourself and the lab's task, and a privacy section if you collect user data or use analytics tools.

3.1 Possible client topics

- Environmental hazard map (Noise pollution, ...)
- Smart City Geosensing
- Geosensing on the move
- Health Risk Commuters

3.2 References

1. **API documentation:**

ArcGIS:

- QuickStart: https://developers.arcgis.com/javascript/latest/guide/index.html
- Examples: https://developers.arcgis.com/javascript/latest/sample-code/index.html
- API: https://developers.arcgis.com/javascript/latest/api-reference/index.html

Openlayers:

- QuickStart: https://openlayers.org/en/latest/doc/quickstart.html
- Examples: http://openlayers.org/en/latest/examples/
- API: http://openlayers.org/en/latest/apidoc/

3.3 W3schools Web Programming Documentation:

- HTML https://www.w3schools.com/html/html_intro.asp
- JS https://www.w3schools.com/js/js_intro.asp
- CSS https://www.w3schools.com/css/default.asp

3.4 Available GeoServices and data sources

Opendata Switzerland	https://opendata.swiss (BGDI WMS)
Stadt Zürich Open Data	https://data.stadt-zuerich.ch/
Kanton Zürich Geodata	https://geolion.zh.ch/geodatensatz
BaseMaps	ArcGIS: Basemap Gallery , Vector Layers Openlayers: Tiled ArcGIS MapServer

4 Requirements for your submission

1. Topic
    a. The web app needs to be tailored to a topic and purpose of your choice.
2. Features:
    a. The web app needs to contain at least 3 different map controls, a description about the web app's target, a short section with a legal notice, an about us element, a privacy notice and a map key (legend). Concerning the legal, about us and privacy notices you can also have a look at the SenseBox website.
3. Code documentation:
    a. Create a Readme file or a header section in the HTML which briefly explains the structure of the web app.
    b. Properly document your code in terms of HTML (<!-- ...--> Tag), CSS (/* ... */) or Javascript comments (//...).
4. Submission:
    a. Publish the URL of your online code editor or open-source learning environment to Moodle (depending on what you chose to use).
    b. Upload a ZIP-file, including your *index.html* file together with all other sources to Moodle. Unzipping and opening *index.html* with Firefox or Google Chrome should run the web app without error on a computer with internet connection (**Please test this before submission! Really! Test it!**).
    c. Please describe the purpose together with your developed tool and its features in a few lines in the Moodle submission form.
5. Remember that your result will be graded. The final grade depends on how much effort you invest in your code and further functionality and on how well your client fits to your stated purpose (see also "Grading").

6. Develop your own client software, do not copy. Solutions that are obviously copied among participants will be judged as plagiarism in the sense of the *Disziplinarordnung ETH* Zürich!

5 Grading

The assignment will be graded according to the following criteria:

- *Correctness and completeness*: to which degree does the interactive web app provide the functionality requested by the requirements?
  - ☐ The SenseBox data is used in a meaningful and comprehensive manner
  - ☐ The Web App has a purpose and a description and reflects its purpose
  - ☐ The Web App runs without error and shows a map
  - ☐ There are at least 3 different map controls
  - ☐ A map key or legend is available
  - ☐ A short section with a legal notice
  - ☐ An about us element
  - ☐ A privacy notice
  - ☐ The code is not copied
- *Code quality*:
  - ☐ The code is easily understandable
  - ☐ Variables are named in a sensible way
  - ☐ The code layout supports the logical structure of the code (e.g., use of indentation, line breaks)
  - ☐ The code avoids unnecessary control structures (e.g., unnecessary loops)
- *Documentation*:
  - ☐ The purpose and structure of the web app is well documented in the HTML header section or in a readme file.
  - ☐ The JS-code is well documented, such that a third party developer should be able to understand what your program is doing at different points of your code (find a balance! Not too much, not too little).

6 Hints using ArcGIS (if you use ArcGIS API for JavaScript)

6.1 MapView (2d) and SceneView (3d)

Quick Overview
https://developers.arcgis.com/javascript/latest/guide/index.html#3d-support
Sample:
https://developers.arcgis.com/javascript//latest/sample-code/intro-sceneview/index.html

6.2 Layers

Introduction to Vector Layers (Feature Layers)
https://developers.arcgis.com/javascript//latest/sample-code/layers-featurelayer/index.html
Introduction to CSV Layer
https://developers.arcgis.com/javascript/latest/api-reference/esri-layers-CSVLayer.html
Sample with a WMS Layer
https://developers.arcgis.com/javascript/latest/sample-code/layers-wms/index.html
OpenStreetMap
https://developers.arcgis.com/javascript/latest/sample-code/sandbox/index.html?sample=layers-osm-3d

6.3 PopUps

Introduction to PopUps
https://developers.arcgis.com/javascript//latest/sample-code/intro-popup/index.html

6.4 3d Rendering (3d Symbols)

Creating visualizations based on numeric values, attributes types with extrusion, opacity, rotation etc.
https://developers.arcgis.com/javascript/latest/guide/creating-visualizations-manually/index.html

Visualizing points with 3D symbols:
https://developers.arcgis.com/javascript/latest/guide/visualizing-points-3d/index.html
API:
https://developers.arcgis.com/javascript/latest/api-reference/esri-symbolsIconSymbol3DLayer.html

6.5 Ready to use Widgets

https://developers.arcgis.com/javascript//latest/sample-code/intro-widgets/index.html

7 Hints using Openlayers (if you use OpenLayers)

7.1 How to request information on WMS layers and their content

http://wms.geo.admin.ch/?SERVICE=WMS&REQUEST=GetCapabilities

7.2 How to use getFeatureInfo in Openlayers

getFeatureInfo allows you to obtain background information about a feature.

For an example on how to use getFeatureInfo look at:

http://openlayers.org/en/latest/examples/getfeatureinfo-tile.html

Attention: You need to check (getCapabilities) first which info format is supported by a service.

7.3 How to add a LayerSwitcher

OpenLayers 3 does not implement yet a LayerSwitcher (i.e., the possibility to, for example, dynamically include or exclude different layers). Alternative, you can use the implementation provided from https://github.com/walkermatt/ol3-layerswitcher.

To include the library to your own website you have to include the following lines to your index.html file:

```
<script src="https://raw.githubusercontent.com/walkermatt/ol3-layerswitcher/master/src/ol3-layerswitcher.js"></script>

<link rel="stylesheet" href="https://rawgit.com/walkermatt/ol3-layerswitcher/master/src/ol3-layerswitcher.css" type="text/css">
```

Afterwards you have to add the next lines in your Javascript:

```
// creating a new group for the layers
var name_of_group_1 = new ol.layer.Group({
        'title': Title_1,
        layers: [first_layer, second_layer, ...]
});

// creating a new group for the other layers
var name_of_group_2 = new ol.layer.Group({
        'title': ' Title_2',
        layers: [first_layer, second_layer, ...]
});

// initializing the layer switcher
var layer_switcher = new ol.control.LayerSwitcher({
        tipLabel: 'Active Layers'
});

// Adding the layers to the map
```

```
map.addLayer(name_of_group_1);
map.addLayer(name_of_group_2);

// Adding the layer switcher to the map
map.addControl(layer_switcher);
```