ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

geoinformation engineering
ikg eth zurich

David Rudi, Christian Sailer

Geosensors with senseBox – Lab 03

Source: **https://sensebox.de/**

Overall Objectives

In this lab, you merge your data with further open context data and visualize them in the web by using a 3d library.

1 This lab: Embed context data and visualize your project in the web

1.1 Goals

- Learn to download meaningful context data of web providers
- Learn to harmonize different datasets into an equal structure and an exchangeable format
- Learn to create a web map service and render geosensor data as a web application using in 3d js components

1.2 Tasks

1.2.1 Recap of last week

🕒 15min 👥 Group Work

Extract and analyse your data (datalog.txt), and verify within peers, the types and values and check if they are plausible.

1.2.2 Search for third party data sources, download and merge them with yours

45min Group Work

In this lab, you will build an ambient sensing application, similar to Opensensemap. Such a Volunteered Geographic Information (VGI) system, gains credibility if the data is supported by (scientific) evidence. Therefore, we want to enrich your data with data from different origins (sensors). The aim is not only to validate your single measurements, but the quality of the application you will develop improves through the additional information about the situational conditions of the surrounding area.

First, have another look at the sensor data providers from last week's lab.

Then, define a meaningful extent for which you want to acquire the information based on the geographical range, the density of measuring stations and the terrain characteristics.

Next, download appropriate data from Opensensemap as well as from other data providers.

Afterwards, clean and harmonize the data structure, that you can later easily compare the content.

Finally, format the data sets in exchangeable open data format. The result should be one aggregated or several individual datasets formatted as CSV.

1.2.3 Upload and publish data on ArcGIS Online

15min Individual Work

As a next step, we publish the CSV data sets on a web server. We use Esri's cloud infrastructure "ArcGIS Online" to do so. You need an Esri Developer Account for that. Please create one at **https://developers.arcgis.com** .

Then, you need to upload your datasets to **https://maps.arcgis.com/home/webmap/viewer.html**, and publish your dataset as vectorised service layer (FeatureLayer). Feature Layers support querying, visualizing, and analysing the data.

> Learn more about the procedures here: **https://doc.arcgis.com/en/arcgis-online/manage-data/publish-features.htm#GUID-DD47C642-38B4-4645-801D-2866FE90BF73** .

Give your service layer an individual style in a web map, which provides you a rich library of classifiers and symbologies. Do not forget to save the style changes in the layers' properties. Finally save the web map for later work.

> Learn more about the procedures here: **https://doc.arcgis.com/en/arcgis-online/create-maps/change-style.htm**

If you want to be independent of locating your data on the web, you can keep the CSV on your own web server. But be aware that if CSV files are not on the same domain as your website, a CORS enabled server or a proxy is required.

1.2.4 Create a web-based map client

45min Individual Work

Programming with ArcGIS:

With ArcGIS Online you can create a web app with a map using a configurable app, the Web AppBuilder, or the Operations Dashboard. These offer you various functionalities, such as layouts and colour schemes, editing and identification tools, social media feeds, side-by-side map viewers, and so on.

Learn more about the procedures here: **https://doc.arcgis.com/en/arcgis-online/create-maps/change-style.htm**

In this lab, the goal is to create a custom solution with an expressive visualization. Therefore, you will develop an application using HTML5, CSS and JavaScript (with the **ArcGIS Javascript API 4.x**).

Feel free to use any online code editor and open-source learning environment of your choice (e.g., **https://codepen.io**, **https://codesandbox.io/**, **http://collabedit.com**). They allow you to create code snippets, test them, and store them und a URL.

For ArcGIS Javascript beginners, follow the sandbox example and further step-by-step online tutorials on **https://developers.arcgis.com/javascript/latest/sample-code/intro-layers/index.html** (there you learn to create a Map, a SceneView and a checkbox input HTML element.)

Map Client Requirements:

Your web app should consist of at least **two sensor data layer** (yours and a third party) which one of them **switchable** (visible / none visible) through a map control (visibility toggles).

Specify the **layer properties** (renderers, opacity, maxScale, etc.), and give data insights in **popups** and **further functionality** of your choice.

Further Hints:

- SceneView **https://developers.arcgis.com/javascript//latest/sample-code/intro-sceneview/index.html**
- Layers **https://developers.arcgis.com/javascript//latest/sample-code/intro-layers/index.html**
- CSV Layer: **https://developers.arcgis.com/javascript/latest/api-reference/esri-layers-CSVLayer.html**
- PopUps **https://developers.arcgis.com/javascript//latest/sample-code/intro-popup/index.html**
- Widgets **https://developers.arcgis.com/javascript//latest/sample-code/intro-widgets/index.html**

Finally, post your map (i.e., the URL to your application) on Moodle.

1.2.5 Resumé on the project

🕒 15min

Review and reflect on the project with your peers. The results are collected in a storymap, which is available at Moodle or at **https://arcg.is/HqGCn**