ELISE – Lot 1: Leveraging the power of location information and technologies to improve Public Services at Local Level: An EU-wide Analysis

Visualization Tool - Installation Guide

Autors:

Sven Casteleyn (Universitat Jaume I)

Aida Monfort Muriach (Universitat Jaume I)

Date:

15/06/2023

Index:

1. Code availability

All code is publicly available via GitHub, at https://github.com/GeoTecINIT/elise.

Using GitHub's standard functionality, the code repository can be forked or cloned, or simply downloaded as a zip file.

2. Operationalising the ELISE Visualization Tool

This installation guide covers operationalizing the ELISE Visualization Tool using GitHub Pages (https://pages.github.com), which was selected as preferred hosting platform under the ELISE Lot 1 project.

The ELISE Visualization Tool can be installed using other hosting platforms that support the basic technologies used (Angular - https://angularjs.org, Node - https://nodejs.org/en/) in a similar way as described in this guide, yet such installation is not covered in this guide.

2.1 Configuring GitHub Pages

To run the ELISE Visualisation tool from Github pages, the latter should be correctly configured to host the website. There are several methods to build and upload to Github Pages, this is the one used in the first released version.

2.2 Download code and modify environment variables

First, **create a fork of the repository** https://github.com/GeoTecINIT/elise under your own organization / username. You now have your own independent repository and you will work in this repository.

Within the code, environment variables (src/environments/) are used to locate the cases.json file in the new Github repository and to build the URL to share the current state of the web app. They need to be updated to reflect your repository.

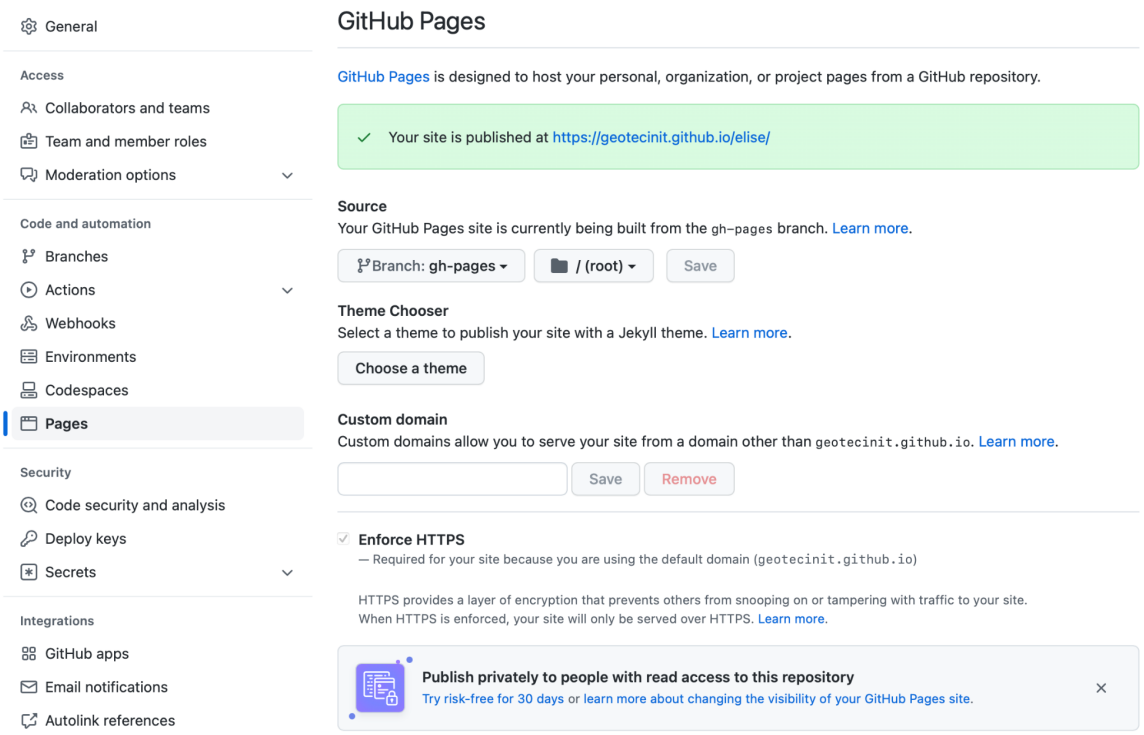
To do so, **substitute the 'base_url' URL** in the files **environment.prod.ts** and **environments.ts**, by the base URL of your repository:

- If your repository was created under an organization, use: https://[ORGANIZATION].github.io/[REPOSITORY_NAME]/#/home
- If your repository was created under your own username, use: https://[USERNAME].github.io/[REPOSITORY_NAME]/#/home,

Theb, **substitute the URL for 'cases_json_url'** in the files **environment.prod.ts** and **environments.ts,** with the new one in the transferred repository. To get this URL, go to src/assets/cases.json and click on the 'Raw' button, then copy the URL.

2.3 Configure Github pages

In the Github website, go to repository settings, and then 'Pages' in the left menu. Select the branch 'gh-pages' as the source for building the web page (click save).



2.4 Generate a personal access token for GitHub Repository

To upload files to a GitHub repository, GitHub is no longer allowing username & password authentication (since August 2021). Instead, a personal access token is now required, which needs to be created in the GitHub repository that requires access.

To create a personal access token, please follow the GitHub documentation (create it for your ELISE repository):

https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

When creating the personal access token, providing full “repo” permissions should be enough (even though further permissions may be required depending on your github setup).

2.5 Build the app and upload it to the gh-branch

For the changes in the environment to be reflected (done in step 2.2), the app needs to be built on your development computer (for example, your local computer) and then this build needs to be uploaded to the gh branch (set up in step 2.3). First, install the necessary tools on your development computer to be able to build the ELISE Visualization Tool:

- Node.js - https://nodejs.org/en/download/ (developed and tested with version 14.16.1)
- Angular CLI: 11.0.4 - https://angular.io/cli (developed and tested with version 11.0.4)
- Git - https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- npm: included with Node.js

Installing more recent versions of the above libraries may lead to warnings or errors while building the project. Once the necessary tools are installed, please follow the following steps (check out the section 2.7 Troubleshooting in case of problems):

1/ From GitHub, clone your ELISE repository to the development computer, and on the development computer, go to the folder of the repository using the terminal, and run the following commands:

2/ Install the app dependencies (warnings may occur when using more recent versions of node.js - 14.16.1 - and angular CLI - 11.0.4):

```
→ npm --force install
```

3/ Install the github pages cli:

```
→ npm i angular-cli-ghpages --save-dev
```

4/ Build the app with the new repository URL:

```
For a repository created under an organization:
→ ng build --prod --base-href "https://[ORGANIZATION].github.io/[REPONAME]/"


For a repository created under your own username:
→ ng build --prod --base-href "https://[USERNAME].github.io/[REPONAME]/"
```

5/ Upload the build to the Github pages (when prompted for username, provide your GitHub username; when prompted for password, provide your personal access token as created in step 2.4 - please see Troubleshooting in case of GitHub login problems):

```
→ npx angular-cli-ghpages --dir=dist
```

Once the upload is successful, you should see the following or similar message:

```
🚀 Uploading via git, please wait...
🌟 Successfully published via angular-cli-ghpages! Have a nice day!
```

2.6 Visit URL and check website works

Please be aware that, after a successful upload, it can take up to 5 - 10 minutes before GitHub Pages is updated and the webpage is visible.

For deployment under an organization: https://[ORGANIZATION].github.io/[REPONAME]/
For deployment under a user account: https://[USERNAME].github.io/[REPONAME]/

2.7 Troubleshooting

1/ JavaScript Heap error

At several of the steps described in 2.5, it is possible you get an error message related to:
→ JavaScript heap out of memory

This error signifies a lack of memory allocated to JavaScript by node.js, due to which JavaScript execution fails (this happens particularly on Windows computers). To solve this issue, you can try to increase the heap size using the following command (and then resume the installation at the step that previously failed):

```
→  set NODE_OPTIONS="--max-old-space-size=8192"
```

In case the error continues to occur, you can try to further increase the heap size, by multiplying the value 8192 by 2 (so the next value would be 16384, 32768 - the max possible value depends on the available memory in your computer).

2/ GitHub login problems

For several reasons, authentication at GitHub may fail in step 2.5.5 above. The most common cause is using a wrong username and/or password. This may inadvertently happen if you are using a credential manager (for example, Windows Credential Manager on Windows; KeyChain on Mac) or a git configuration (for example, in a config file such as .gitconfig, .git/config, ~/.git-credentials or any similar config file) which has different/wrong username/password configured (remember, the personal access token needs to be used as password - see step 2.4!).

Next to solving the configuration issues with your credential manager or git, in step 2.5.5, you can also try to manually set the upload repository and use your personal access token, as follows:

```
For a repository created under an organization:
→ npx angular-cli-ghpages --repo=https://[GITHUB PERSONAL ACCESS
TOKEN]@github.com/[ORGANIZATION]/[REPONAME].git


For a repository created under your own username:

→ npx angular-cli-ghpages --repo=https://[GITHUB PERSONAL ACCESS
TOKEN]@github.com/[USERNAME]/[REPONAME].git
```

3. Credits

All code was developed by the GEOTEC research group (http://geotec.uji.es/) of the Universidad Jaime I (http://www.uji.es/), Castellón, Spain, as part of the European Location Interoperability Solutions for e-Government (ELISE) Lot 1 Project carried out for the European Commission's Joint Research Centre (JRC).