GEOTAG INSTALLATION

-MANUAL-

GEOSYS

Index

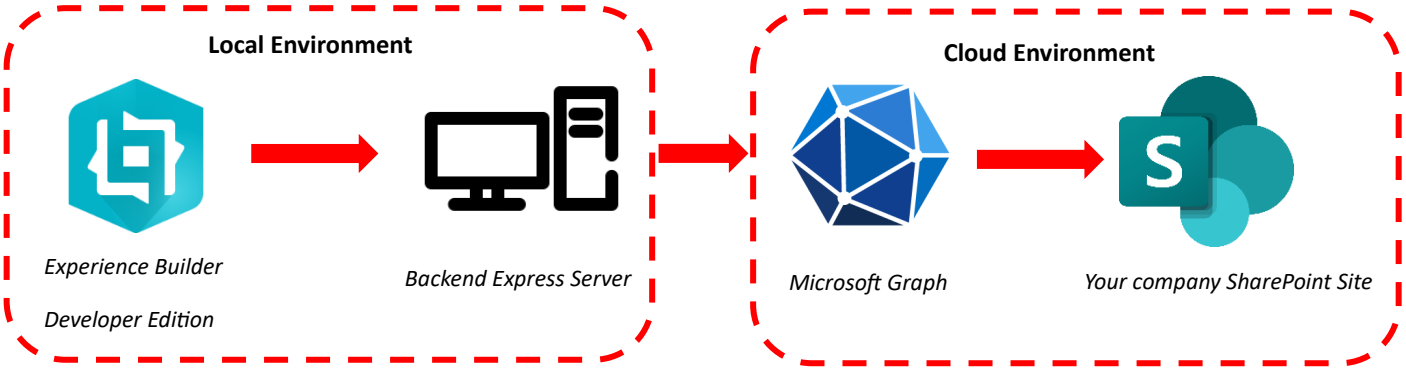
Introduction

We Geosys have developed a powerful custom widget for Experience Builder, capable of seamlessly browse files and folders stored on your company's SharePoint Sites, pinpoint them on a map and later retrieve them using a user-friendly interface integrated within Experience Builder, merging these two powerful worlds of storing data and GIS.

In this manual you will find an explanation of the technologies that we used for a better understanding of the widget, and how to install it and configure it.

If your company already installed the widget, please refer to the other manual, "GeoTag Usage" to know how to use it and take full advantage of.

Clarified that, this is a schema of all the subprocess that the Widget goes through:



As we indicate on the schema, there are two services that need to be installed locally. While we recommend using a server machine, It's also functional on a standard computer. For this manual, we demonstrate using a Windows Server machine, but the steps are equally applicable to a regular Windows machine.

For a more detailed understanding of how the widget operates, please refer to the "GeoTagFlowchart" image in the Documentation folder.

About the technologies

Experience Builder Developer Edition

For having custom widgets on Experience Builder, Esri gives plenty of tools to work with, but also there are some limitations that we are going to point out. The most important one is that currently, on version 1.12, there is no way to use custom Widgets on Experience Builder Online. Esri has a specific edition for testing and using those custom Widgets called “Experience Builder Developer Edition” that visually and functionality is identical to EB Online.

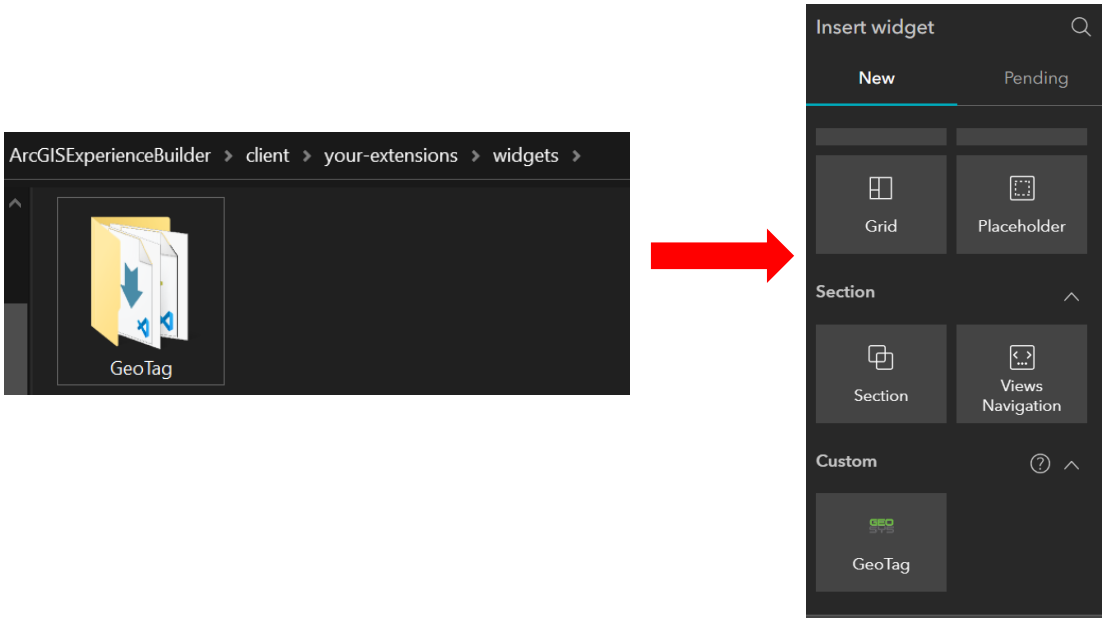
EBDE (for now on, we will use this short name for referring to Experience Builder Developer Edition) works locally on a server, but you can easily deploy the Web Apps that you will create to ArcGIS Online or Enterprise. Also, if you have existing “experiences” on your EB Online or in your computer you can import them onto the Dev Edition easily.

EBDE SDK has two folders, the “server” and “client”. Both are different services that are executed separately, but they need each other to work properly.

By default, it runs on port 3001 with the HTTPS protocol using a self-signed certificate.

Another thing to consider about EBDE is that when Esri updates the Experience Builder to a new version, you need to manually download the SDK with the latest version and move the custom widgets to the new version. If you don’t have the latest version of EBDE, it can cause some problems like, maybe if you want to import an experience from EB Online maybe it will be incompatible, so make sure always to have the latest version. You can check the latest versions of Experience Builder on here: https://developers.arcgis.com/experience-builder/guide/release-versions/

Here we can see how it works adding a custom widget and how it appears in EBDE:

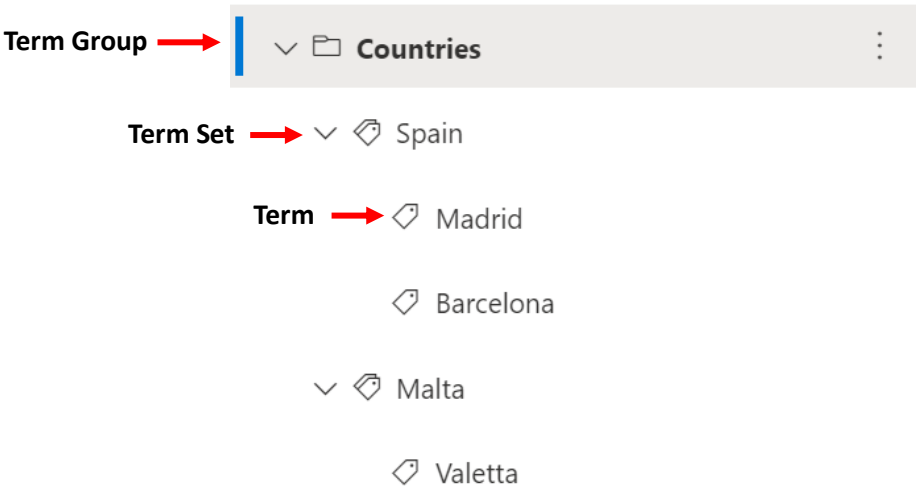


Microsoft Graph

Microsoft has a complex environment, with multiple applications such as SharePoint, Outlook, Office, Teams… Each one has their respective APIs to work with, but Microsoft Graph what it does is grouping all these APIs onto a same Endpoint, where you can access data from SharePoint and Teams and office with ease. For the case of our widget, we only use the features that interact with SharePoint, but is a powerful API that each day is growing.

SharePoint Taxonomy

Taxonomy is the science of classification, and SharePoint gives us a powerful and easy way to classify our files called "Term Store". SharePoint has 3 levels of classification: Term Groups, Term Sets and Terms. The Terms are the identifiers that you apply to files, so you can have a structure like this:

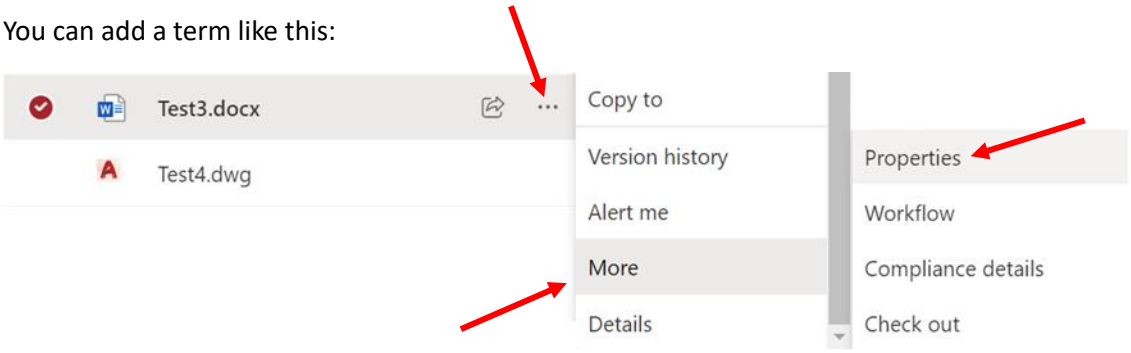


If you want to apply those terms onto files, you will need to create a column that will store those values for all the files of your SharePoint Site (folders cannot contain terms).

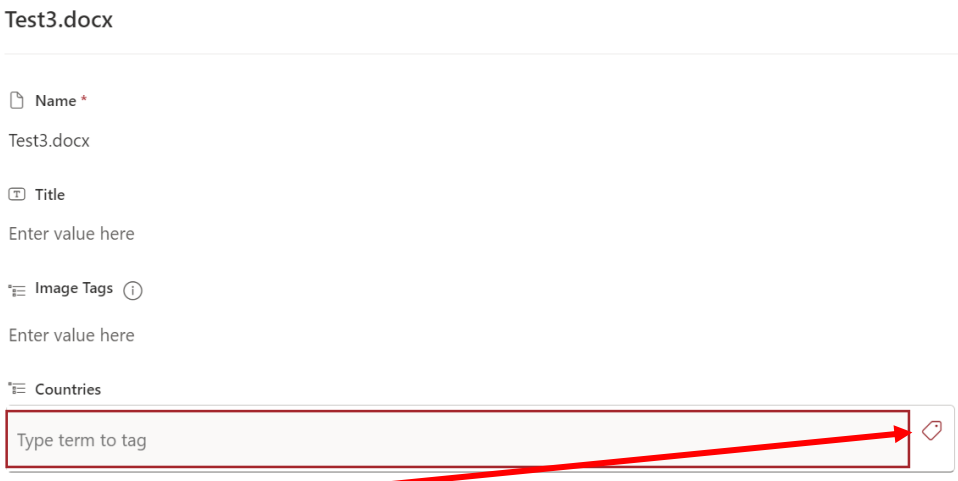
The column can hold a Term Set, and you can apply all the terms of the selected Term Set to the files. Here the column is called "Countries":

Name	Countries	Modified
Test1.docx	Barcelona	A few seconds ago
Test2.xlsx	Madrid	A few seconds ago

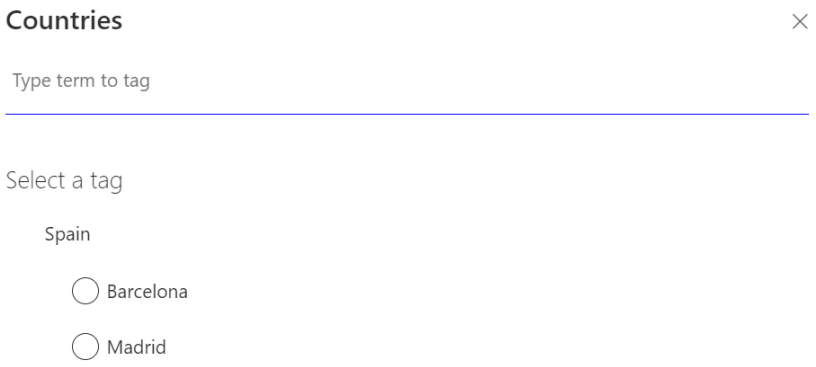
You can add a term like this:



Here it will appear an input where you can write the desired term (it needs to exist the term).

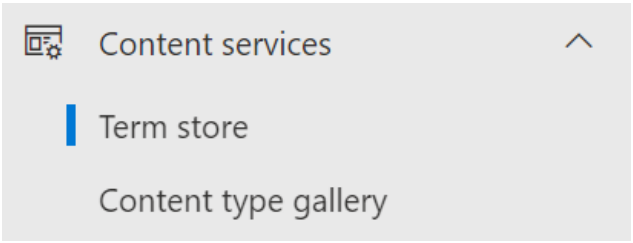


Clicking here will display all the terms that you can apply to a file:



For our Widget, we take profit on this, having a Term Set that stores all the terms that will have the files.

You can find the Term Store on the SharePoint Admin center (only for users with specific roles), inside Content Services

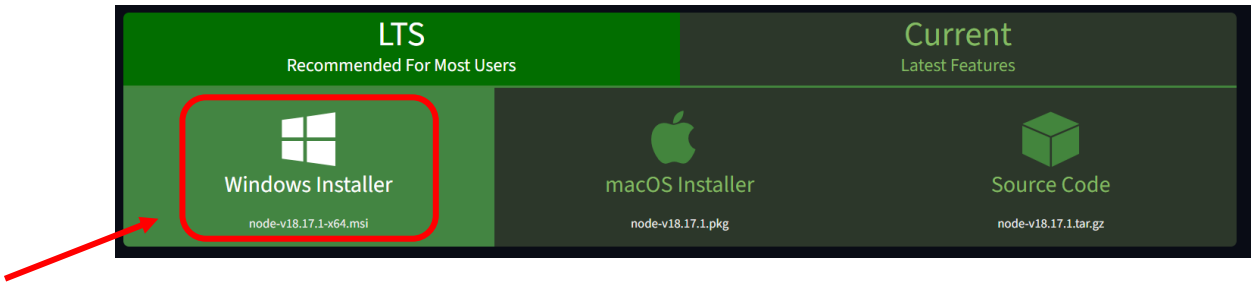


Installation

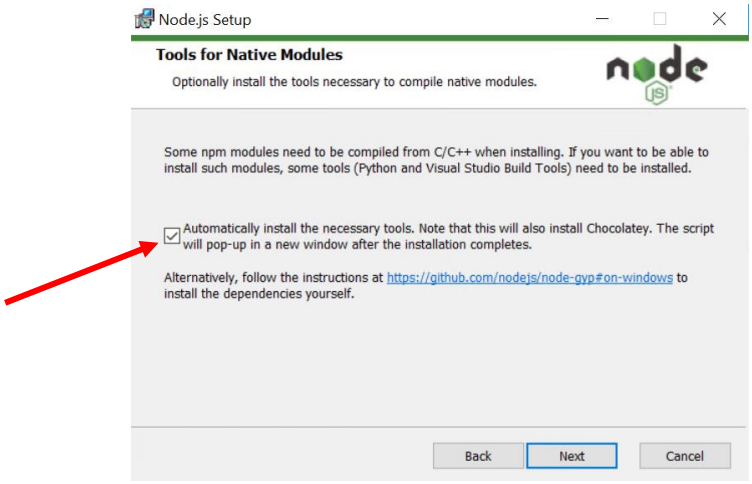
ArcGIS Experience Builder Developer Edition

We are going to explain all the steps for having a functional EBDE server. When making this manual, the latest version was 1.12, so please, also refer to the official documentation for avoiding version issues: https://developers.arcgis.com/experience-builder/guide/install-guide/

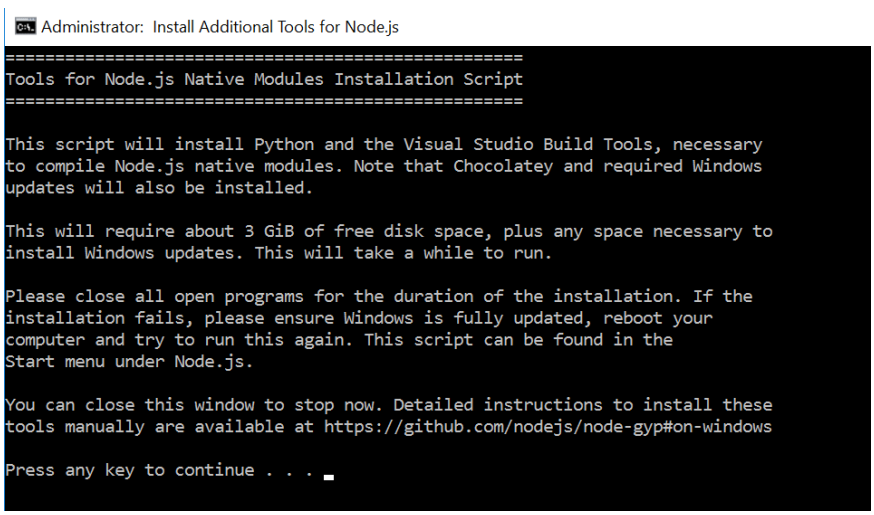
EBDE uses Node.js, when making this manual we were using version 18.17.1, but please, use the latest LTS version for Windows. You can find it here: https://nodejs.org/en/download



The installation is very straightforward, you will only need to select “next” until the next screen, here you will need to check the box for installing all the necessary tools:



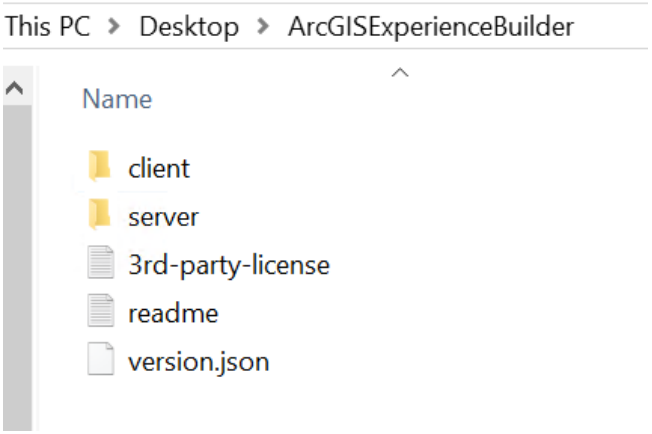
After ending the installation, it will appear a terminal, we just simply need to press any key to continue. This will take a little bit of time.



After the installation ends successfully, we can start downloading the EBDE. Here we can find the latest versions, please download the latest release:

https://developers.arcgis.com/downloads#arcgis-experience-builder

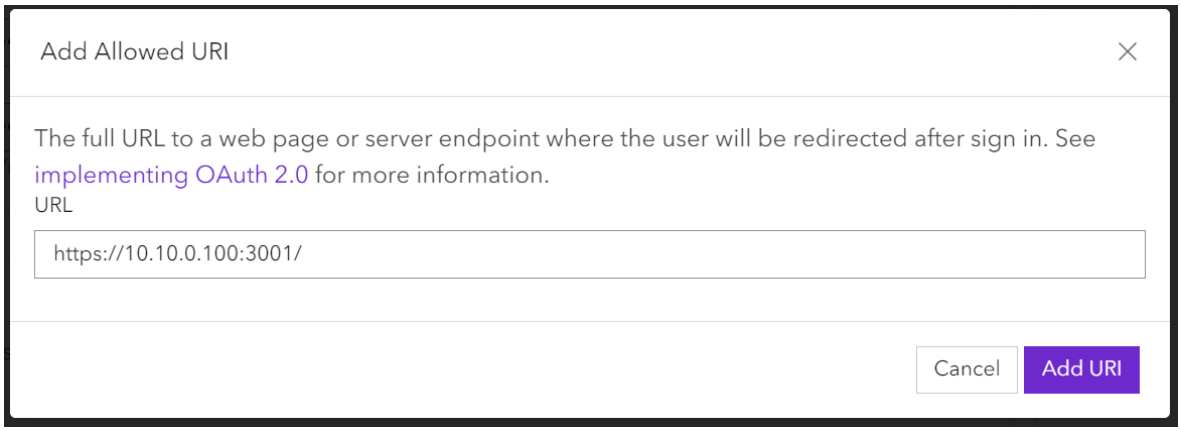
Move it to an accessible folder and unzip it. You will have a file structure like this. The main logic of EBDE is split into two folders: “client” and “server”.



Now you will need to create an OAuth2 App for synchronizing the EBDE with your Esri infrastructure. Go to https://developers.arcgis.com/dashboard/ and click the OAuth 2.0 tab.

Go to new application, give it a name (description not necessary) and after that you will have created the OAuth app. Now you will need to scroll down to the section “Redirect URLs” and click on “Add URL”.

Here you will need to write the IP or name of the machine of your own server, it is important to use https and port 3001. In our case we used the IP of our server that is “10.10.0.100”.

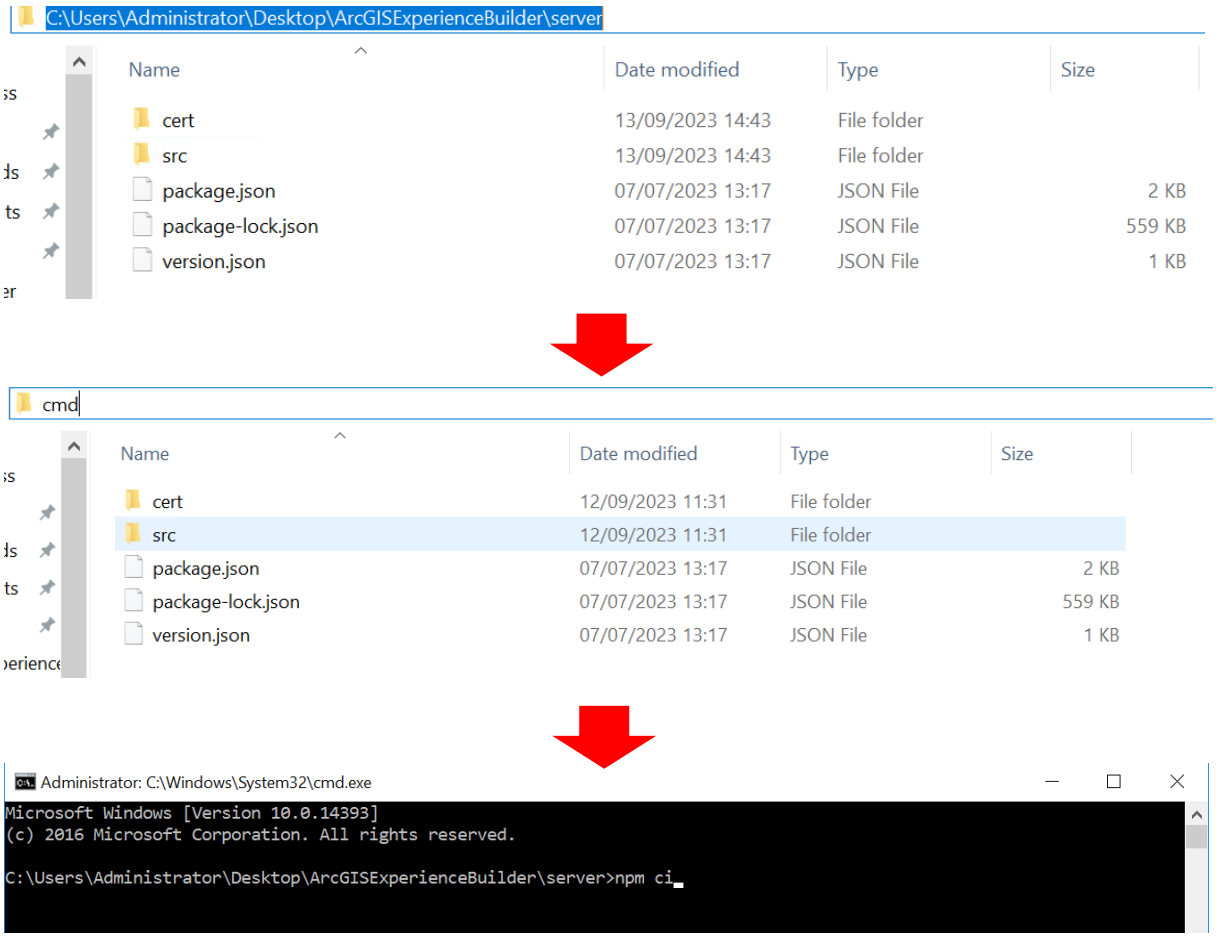


Save the Client ID. All the users will need it for accessing the server for the first time.

Now you will need to install some node packages on the server. With a terminal, you will need to execute the command “npm ci” inside the “server” folder.

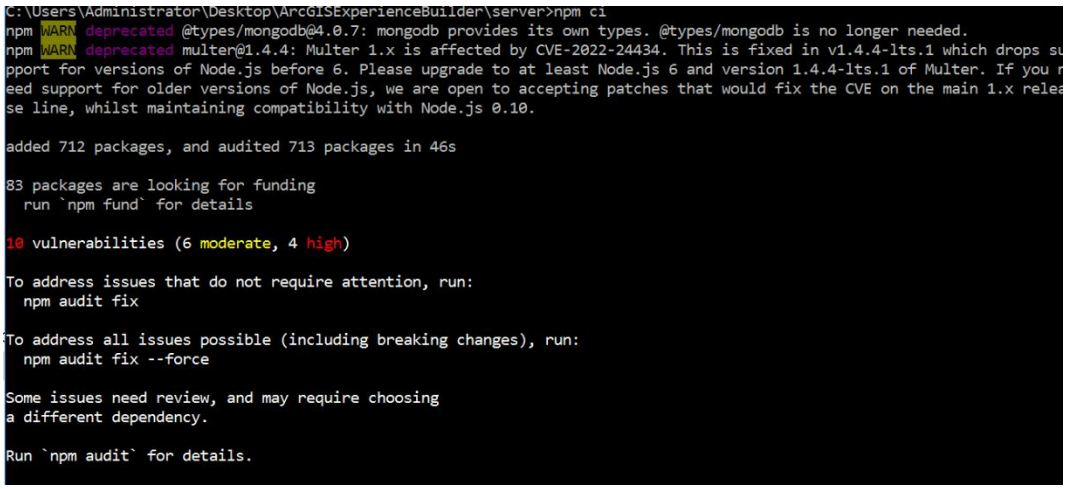
Tip: To open a command prompt directly on the folder that you want you just simply need to write “cmd” on the directory bar:

Delete the current route and just write “cmd”, then press enter:



Write the command “npm ci” and press enter.

Maybe there are some warnings and vulnerabilities, this is common for a node installation, but don’t panic, this will not affect the security or affect the server in some way.

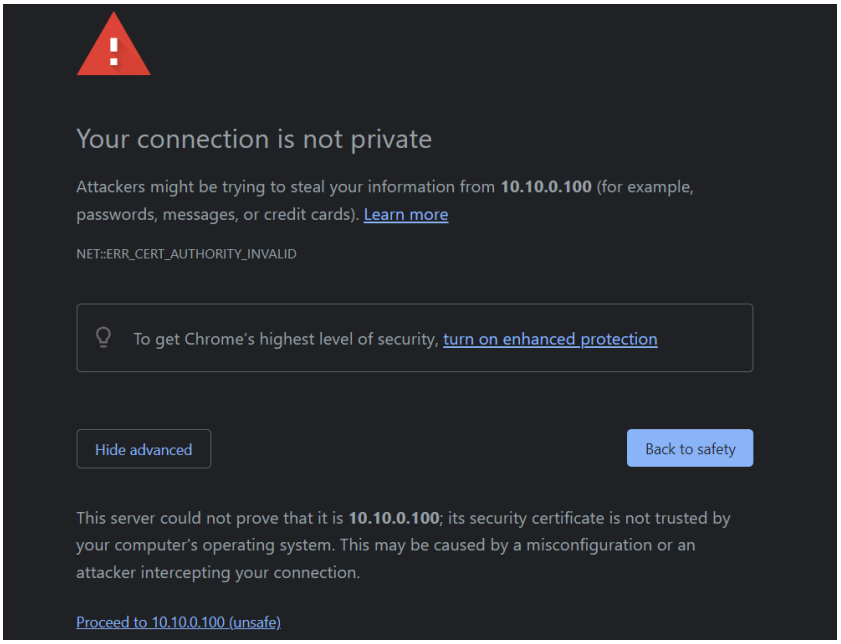


Now you will have finished the installation of the server. Check if everything is working fine by executing on the same terminal “npm start”. This command line is for starting the EBDE manually. If everything is going perfectly, it should appear something like this:

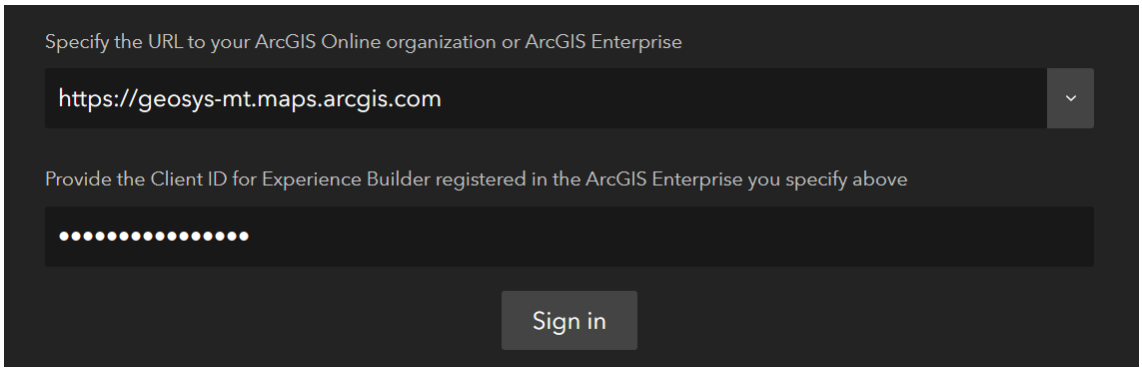
```
C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\server>npm start

> exb-server@0.1.0 start
> cross-env NODE_ENV=production node src/server

Apps folder: C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\server\public\apps
Http server running on port 3000
Https server running on port 3001
```

Try to connect with another computer, using the URL that you specified. If it doesn’t work, check the firewall of your server and if necessary, add an Inbound and Outbound Rule for the port 3001 and 3002 (for the backend server that you will install).

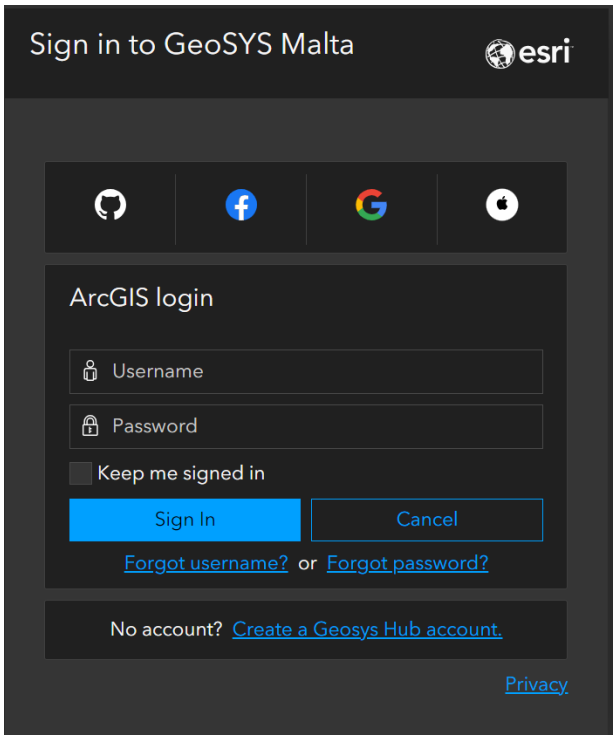
If everything is well configured, it should appear a screen like this indicating that the connection is not private. This is because EBDE is using a self-signed certificate. You can feel free to use your own certificates. But for the moment you will just need to select the option “Proceed to 10.10.0.100”



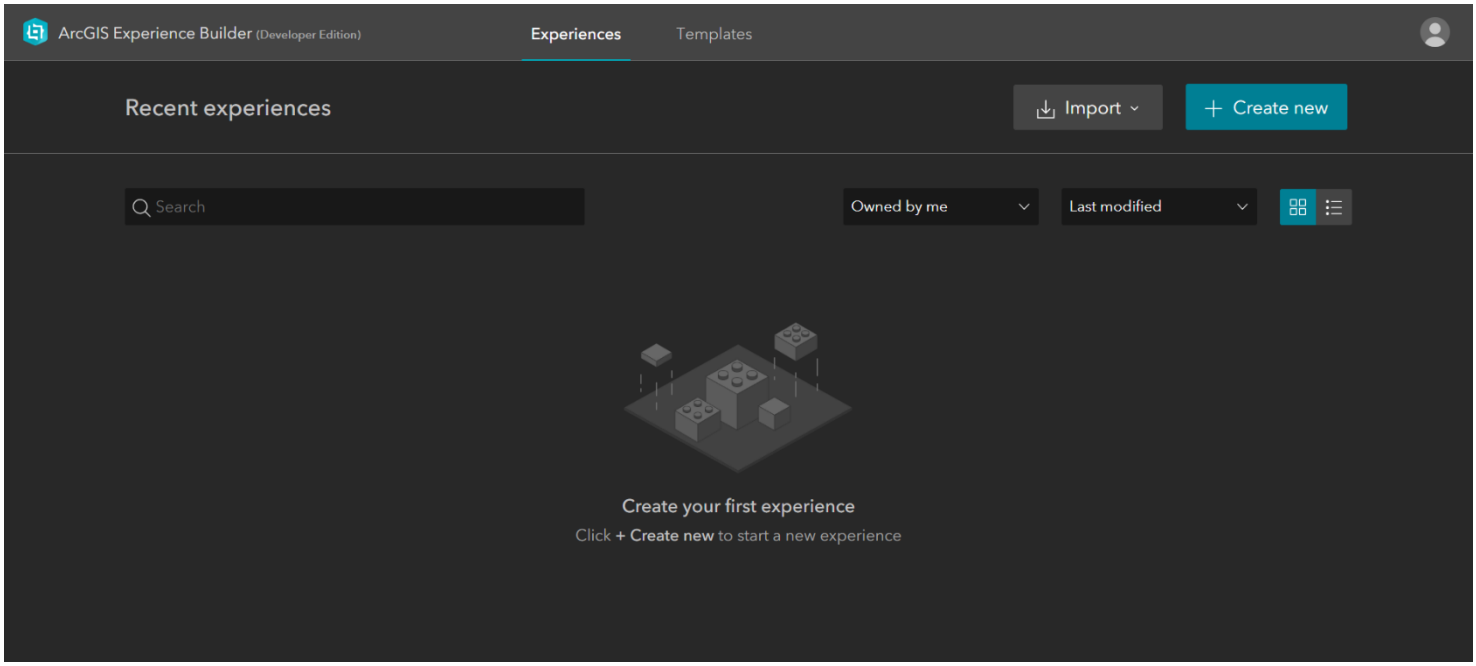
Now you will need to specify your ArcGIS URL organization and the Client ID that we generated before.

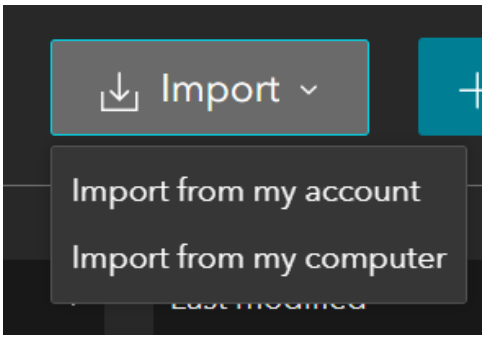


Then it will redirect to a normal esri Sign in page, write your credentials.



And now it should open the main page of EBDE. As you can see, is almost identical to EB Online, with the difference that we have the option of importing our experiences from our account or computer.





Now you will need to go back to the server machine to finish the configuration, this time inside the "client" folder. The client folder is the one that will store the custom widget, for the moment we are going to install it by default, and later we are going to add the widget.

We just need to open a terminal and execute the command "npm ci" again (remember the tip for opening the terminal on the desired folder). It will also show some warnings and vulnerabilities, but nothing to worry about.

```
C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\client>npm ci
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: react-popper@1.3.11
npm WARN Found: react@18.2.0
npm WARN node_modules/react
npm WARN   react@"^18.2.0" from the root project
npm WARN   36 more (@emotion/react, @emotion/styled, ...)
npm WARN
npm WARN Could not resolve dependency:
npm WARN peer react@"0.14.x || ^15.0.0 || ^16.0.0 || ^17.0.0" from react-popper@1.3.11
npm WARN node_modules/reactstrap/node_modules/react-popper
npm WARN   react-popper@"^1.3.6" from reactstrap@8.10.1
npm WARN   node_modules/reactstrap
npm WARN
npm WARN Conflicting peer dependency: react@17.0.2
npm WARN node_modules/react
npm WARN   peer react@"0.14.x || ^15.0.0 || ^16.0.0 || ^17.0.0" from react-popper@1.3.11
npm WARN   node_modules/reactstrap/node_modules/react-popper
npm WARN     react-popper@"^1.3.6" from reactstrap@8.10.1
npm WARN     node_modules/reactstrap
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated @npmcli/move-file@2.0.1: This functionality has been moved to @npmcli/fs
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use
 @rollup/plugin-terser
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with
15x fewer dependencies
```

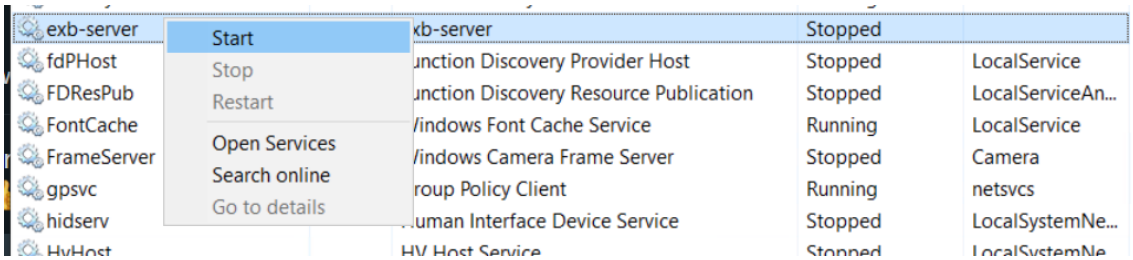
If we want to start it manually, we can also use "npm start", but for the moment is not necessary because we haven't added the widget yet.

Now the last part is making the server folder as a Windows service. We must go again to the command prompt inside the "server" folder. Before that if you still have the "server" or "client" started (with the command npm start that you used before) you need to stop it by pressing "Ctrl+C". Caution, closing the command prompt doesn't stop the service, so be careful with that.

After being sure that nothing is running, we need to open the terminal inside "server" and write "npm run install-windows-service". If everything went fine, it should appear the following:

```
C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\server>npm run install-windows-service

> exb-server@0.1.0 install-windows-service
> winser -i --startcmd="node src/server"

Use start command "node src/server".
The program "exb-server" was installed as a service.

C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\server>
```

Now we need to open the task manager and then go to “Services”. There we will need to find the service “exb-server” and we need to Start it.

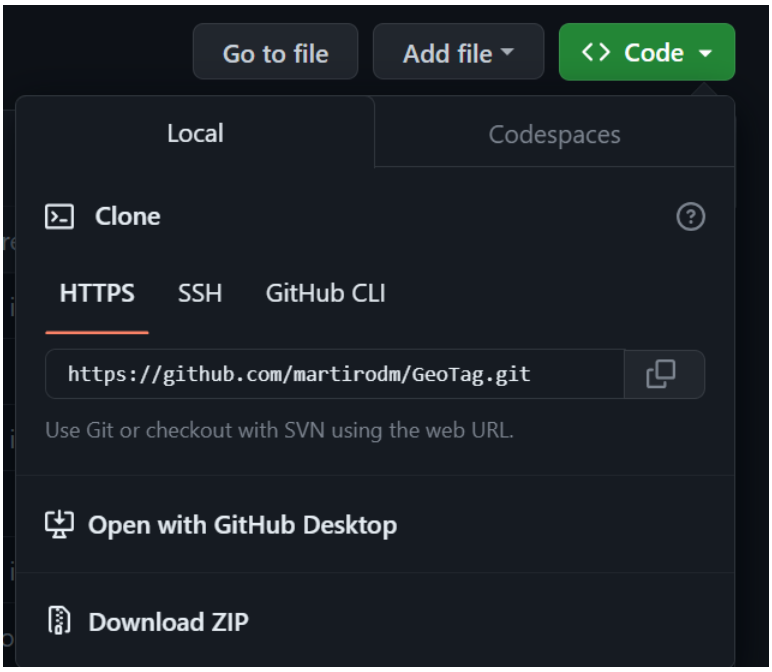


Now the EBDE is a service, so this means you will not need no write each time you want to start the server “npm start” because it will execute with the server automatically.

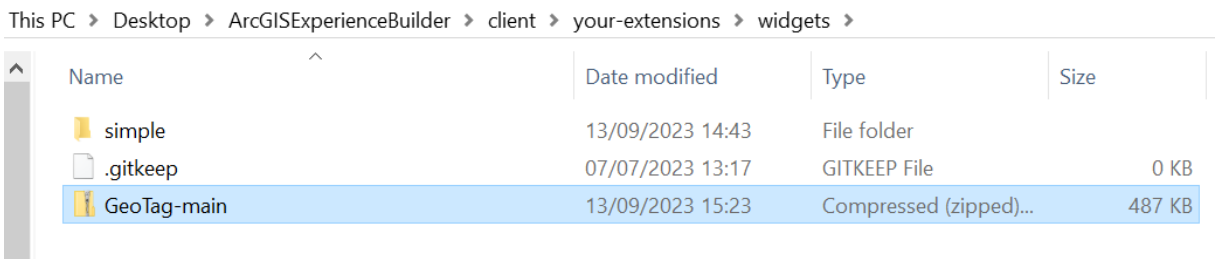
GeoTag Widget

Now you will need to add the custom widget that we GEOSYS, developed. Firstly, go to the next Github Repository: https://github.com/martirodm/GeoTag

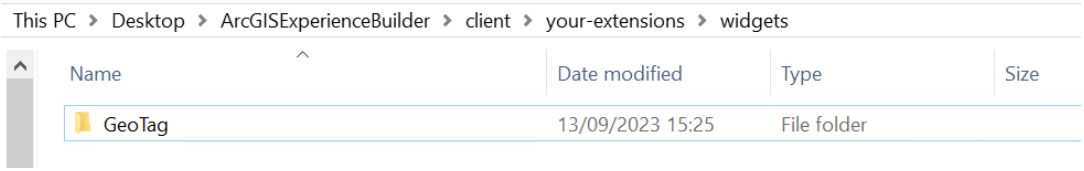
Then if you go to <> Code you will se some options to download or clone the code. If you have .git on your computer you can simply clone it, and if it is not the case just download the ZIP



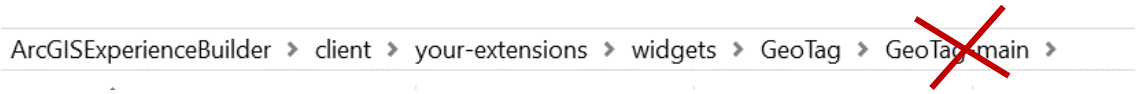
You will need to clone it or decompress the zip on the folder “widgets” that is inside “your-extensions” on the folder “client”. You can delete the file “.gitkeep” and the folder “simple”.



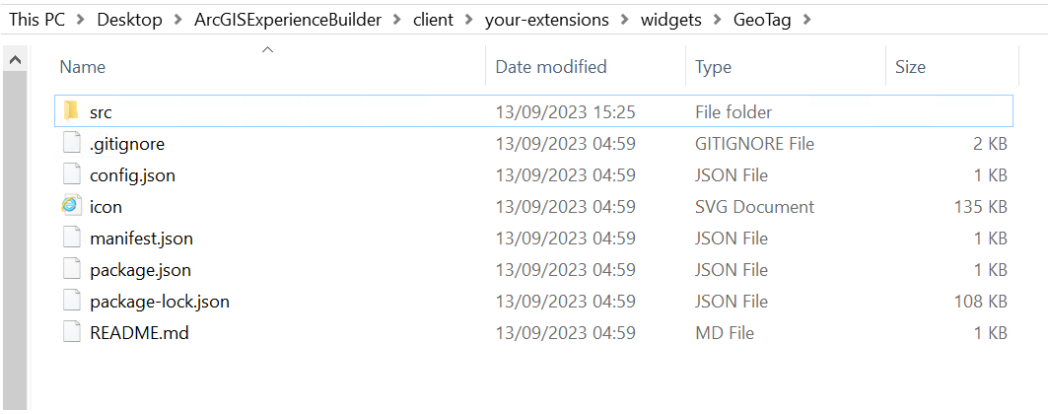
After extracting the zip, delete the zip and only leave the folder. Now change the name of the folder to only “GeoTag”. This step is very important, and it needs to be exactly identical the name.



Also, if inside GeoTag there is again the same folder that contains all the files, just move all the files inside the first GeoTag folder and delete the other folder.

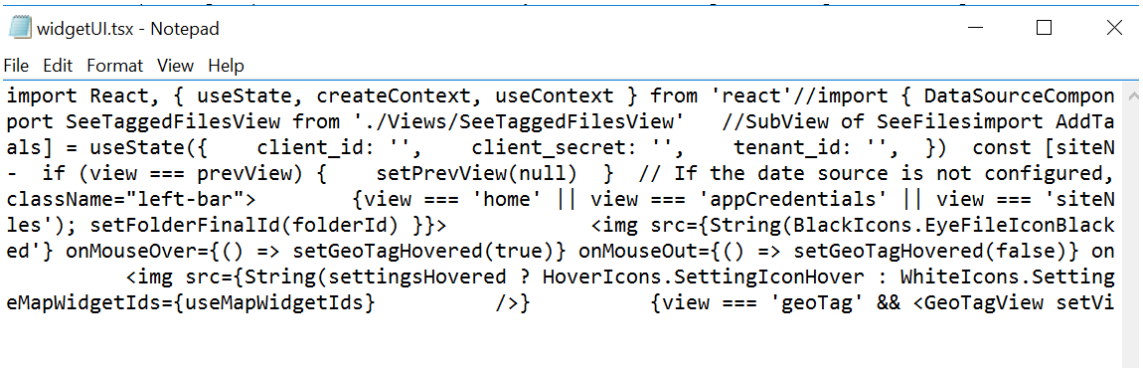


It should be like this:

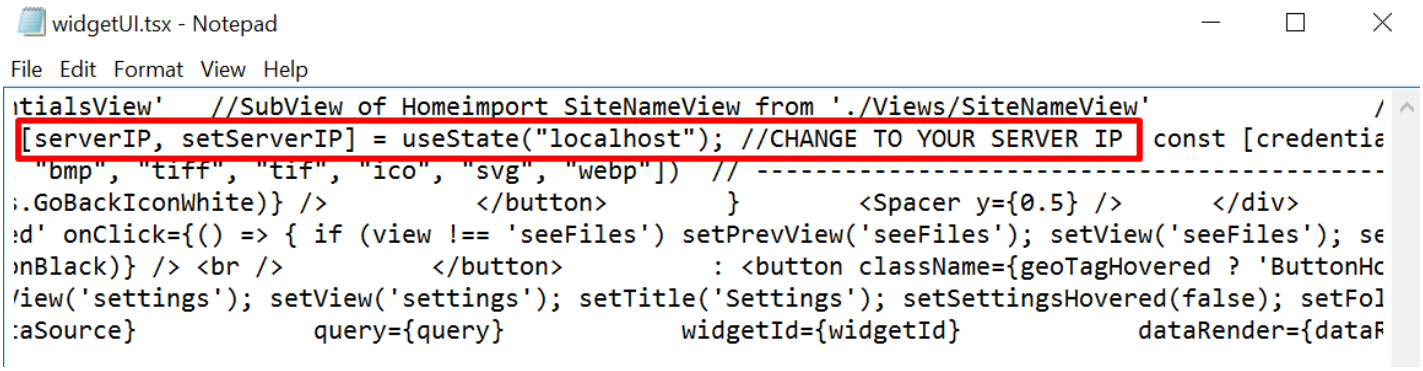


You will need to change some parameters. Go to “src”, then to “runtime” and open the file “widgetUI.tsx”, you can use notepad if you don’t have any code editor.

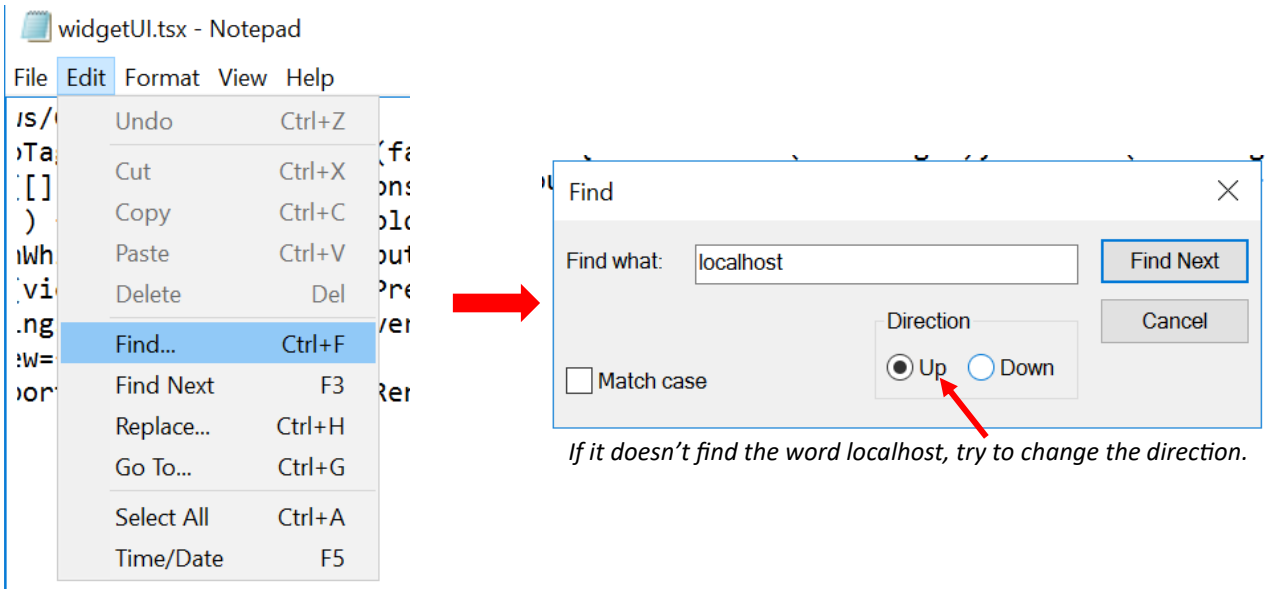
It will look something like this:



You need to find the following piece of code:

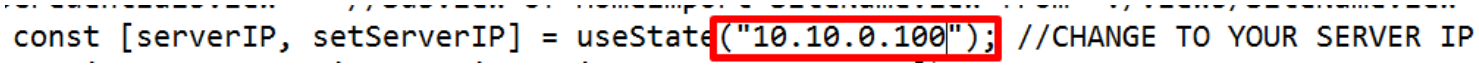


If you don't find it, you can use the search tool of notepad:



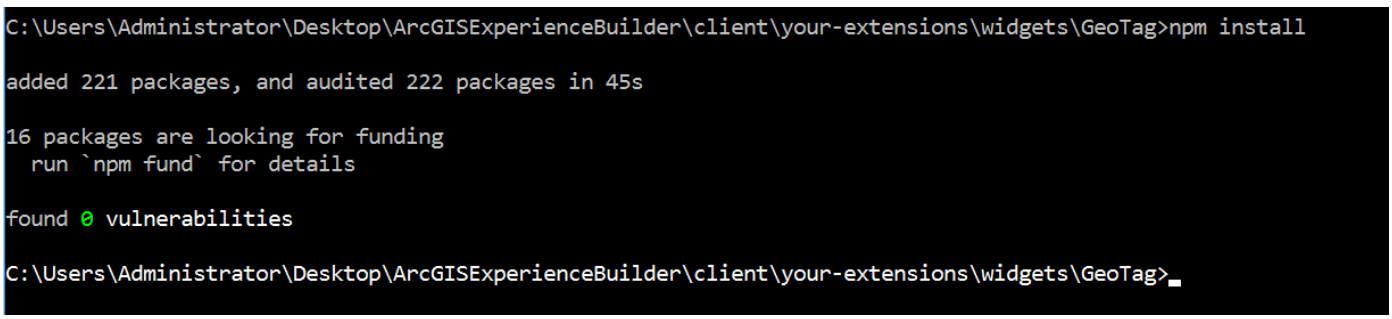
If it doesn't find the word localhost, try to change the direction.

Now you will need to replace the word "localhost" with the IP of you server machine.



Save and close the notepad.

After that, inside the folder GeoTag (be careful that has all the files, not the duplicated folder), open a command prompt and execute "npm install".

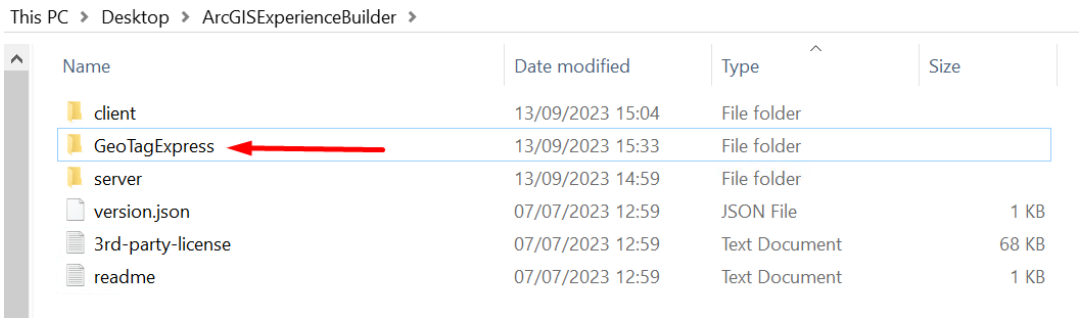


Close the terminal after that.

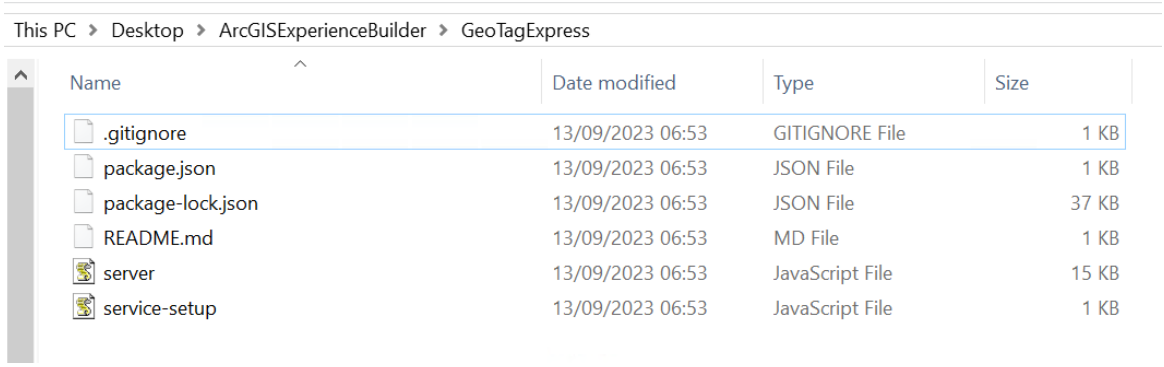
Now you will need to download a Backend server that will make the connection between EBDE and Microsoft Graph. You will find the backend server on the next github:

https://github.com/martirodm/GeoTagExpress

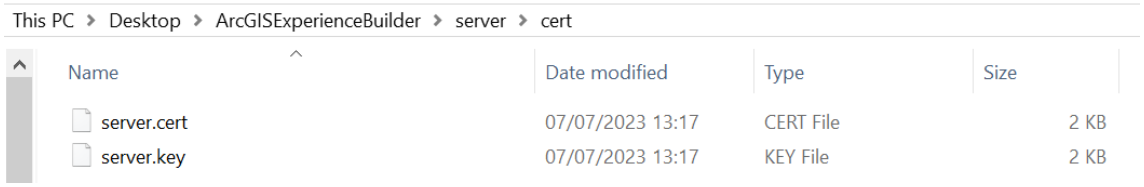
Clone the project or download the Zip onto the folder “arcgis-experience-builder-version”. Extract it and change the name to “GeoTagExpress”:



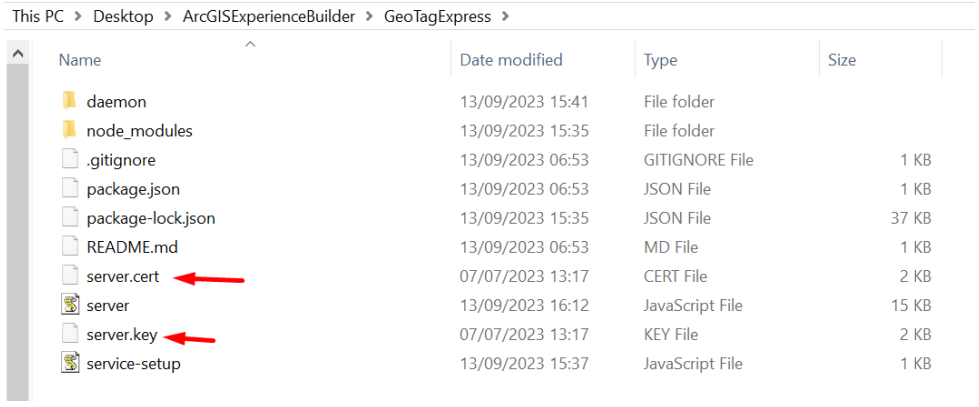
If it happens again the problem of having the same folder inside, just do the same, move all the files to the first folder and delete the duplicated one.



After that you will need to copy the self-signed certificate of the server and paste it on the GeoTagExpress folder. You will find the certificates here:



Just copy-paste it to the GeoTagExpress folder



Now open inside here again a command prompt and write the command “npm install”:

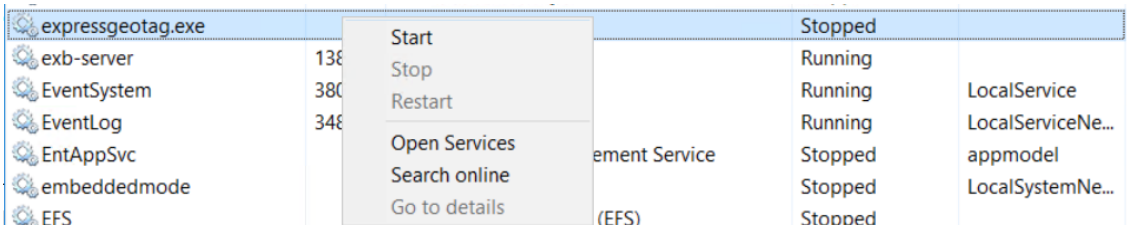
```
C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\GeoTagExpress>npm install

added 97 packages, and audited 98 packages in 4s

13 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\GeoTagExpress>
```

After that, you will convert it to a Windows Service, like we did before with the “server” folder. Now write the command “node service-setup.js”

If everything went well it should say “Service installed”.

```
    description: 'ExpressGEOTAG',
    logpath: null,
    env: undefined,
    execPath: null,
    logOnAs: {
      account: null,
      password: null,
      domain: 'WIN-D4JBFT9UD43',
      mungeCredentialsAfterInstall: true
    },
    workingdirectory: 'C:\\Users\\Administrator\\Desktop\\ArcGISExperienceBuilder\\GeoTagExpress',
    stopparentfirst: undefined,
    stoptimeout: 30,
    logmode: 'rotate',
    logging: undefined,
    allowServiceLogon: undefined
  }
}
Service installed.
```

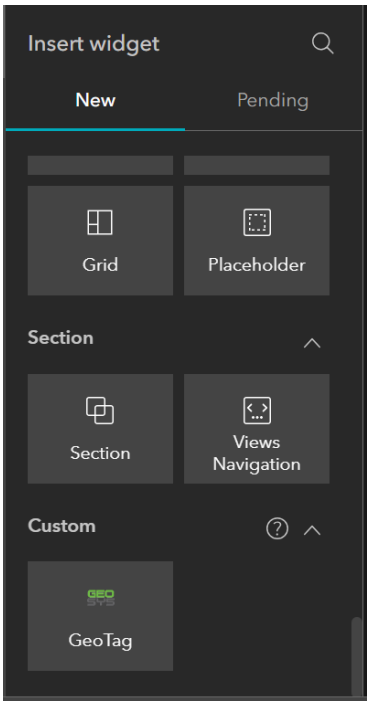
Check on task manager to see the service. Is called “expressgeotag.exe”. If it’s stopped, start it.



Now that we have the two services started, we just need to activate the widget. On the “client” folder execute “npm run build:prod”. It would appear some warnings, but it doesn’t affect the widget.

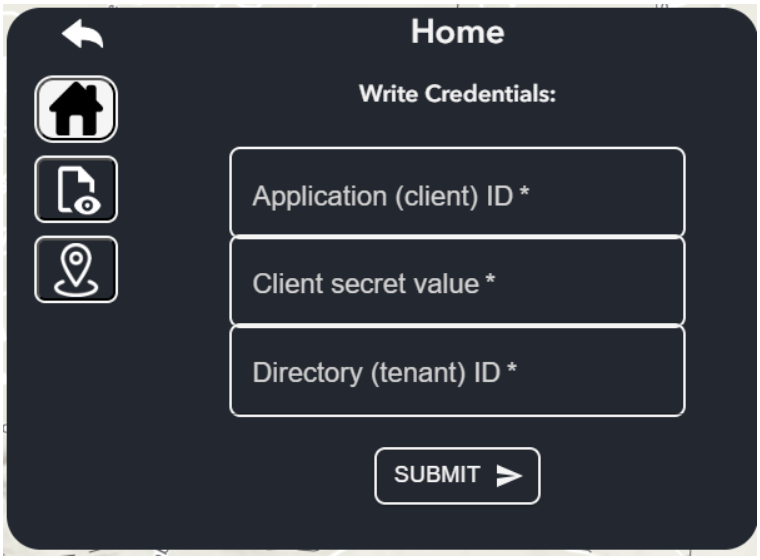
```
C:\Users\Administrator\Desktop\ArcGISExperienceBuilder\client>npm run build:prod

> exb-client@1.12.0 build:prod
> cross-env NODE_ENV=production OUTPUT_FOLDER=./dist-prod webpack --mode production

Entrypoint widgets/GeoTag/dist/runtime/widget [big] 500 KiB = widgets/GeoTag/dist/runtime/widget.js
Entrypoint widgets/GeoTag/dist/setting/setting 2.03 KiB = widgets/GeoTag/dist/setting/setting.js
chunk (runtime: widgets/GeoTag/dist/setting/setting) widgets/GeoTag/dist/setting/setting.js (widgets/GeoTag/dist/setting/setting) 2.67 KiB (javascript) 697 by
tes (runtime) [entry] [rendered]
  runtime modules 697 bytes 4 modules
  dependent modules 126 bytes [dependent] 3 modules
  cacheable modules 2.55 KiB
    ./jimu-core/lib/set-public-path.ts 331 bytes [built] [code generated]
    ./your-extensions/widgets/GeoTag/src/setting/setting.tsx 2.23 KiB [built] [code generated]
chunk (runtime: widgets/GeoTag/dist/runtime/widget) widgets/GeoTag/dist/runtime/widget.js (widgets/GeoTag/dist/runtime/widget) 1.1 MiB (javascript) 999 bytes
(runtime) [entry] [rendered]
  dependent modules 428 KiB [dependent] 112 modules
  runtime modules 999 bytes 6 modules
  cacheable modules 698 KiB
    ./jimu-core/lib/set-public-path.ts 331 bytes [built] [code generated]
    ./your-extensions/widgets/GeoTag/src/runtime/widget.tsx + 209 modules 698 KiB [built] [code generated]

WARNING in asset size limit: The following asset(s) exceed the recommended size limit (244 KiB).
This can impact web performance.
Assets:
  widgets/GeoTag/dist/runtime/widget.js (500 KiB)
```

So now if we go to https://your-serverip:3001/ and work on an experience, it should appear the widget:

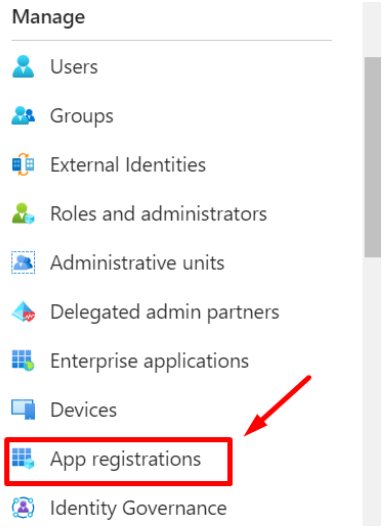


Microsoft Graph and Sharepoint configuration

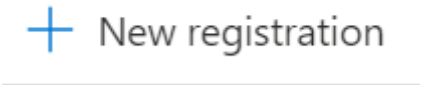
The widget asks you for some credentials, this are for syncing the SharePoint of your company with the Widget. Let's see how we can get these values. You need to be a Microsoft Global Admin for doing these steps.



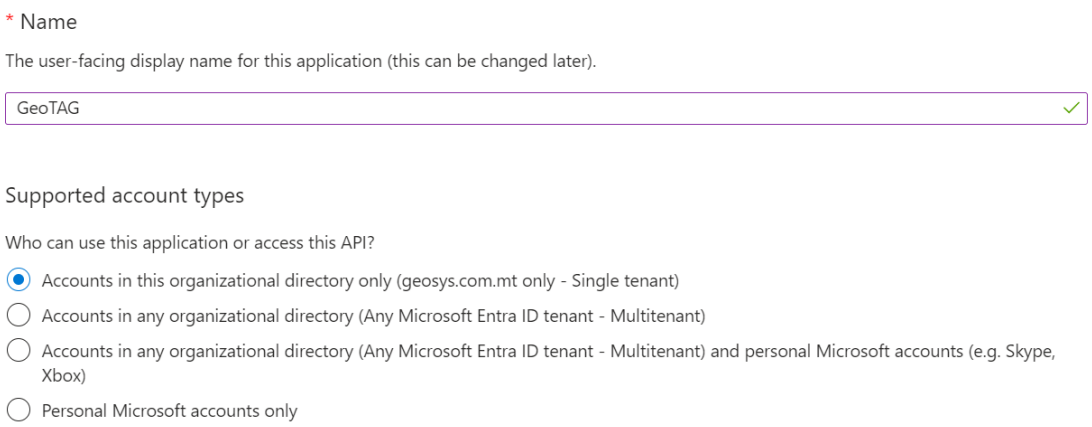
Firstly, you will need to open the portal of Azure: https://portal.azure.com/ and go to the Azure Active Directory. Once there you will need to go to "App registrations"



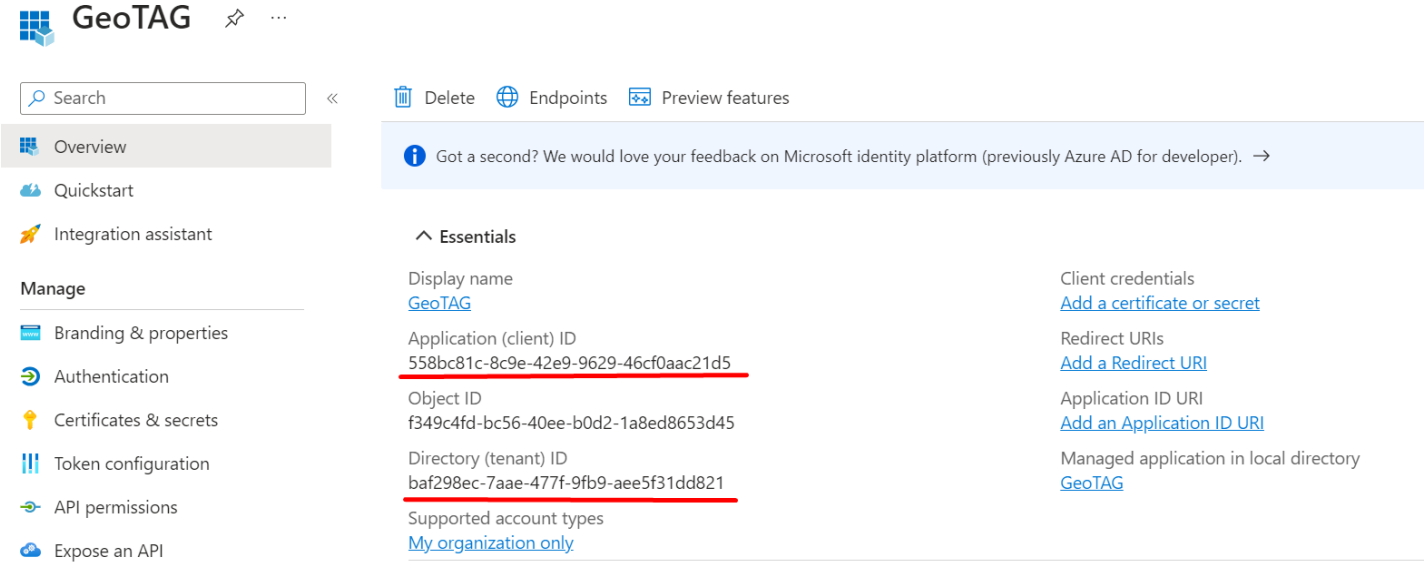
And create a new registration



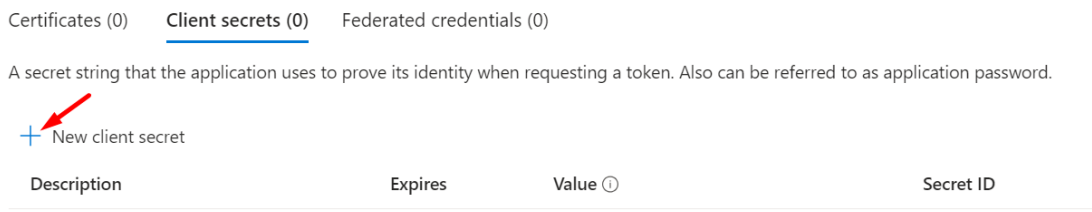
You can name it as you want, but you need to select the option “Single tenant”. Ignore the “Redirect URI” option



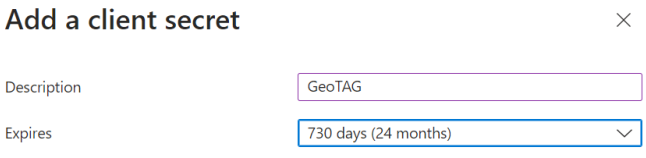
Now, we have created the app. Here we can find two of the 3 required values for login on the widget.



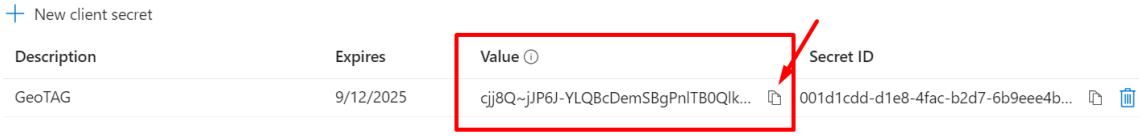
We can get the last value on “Certificates & secrets”. Click on new client secret



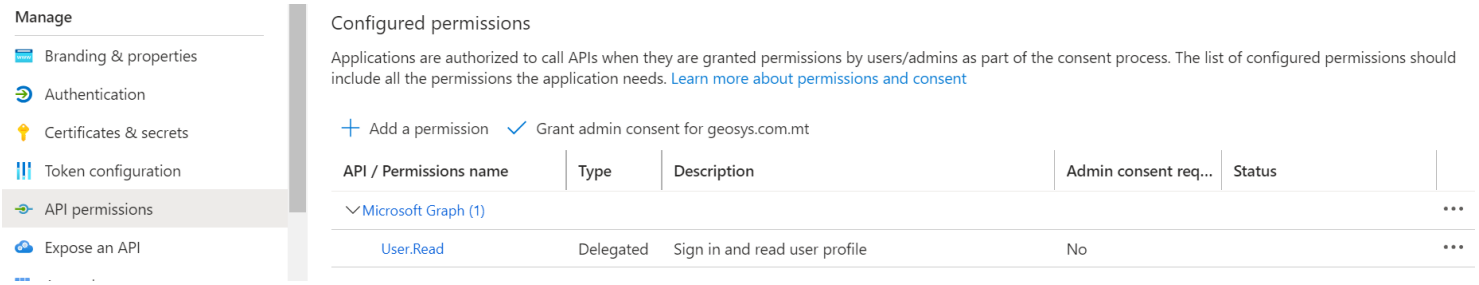
Then write the description that you want a choose an expiration date. We recommend 24 months.



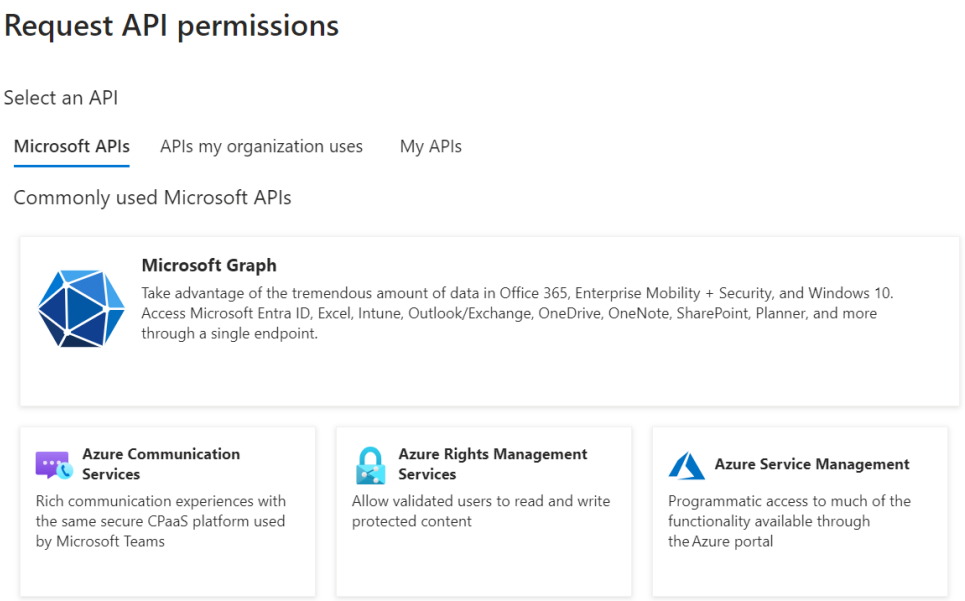
Copy the value. Is important to save this key on a safe place, as the next time you reload the page you will never see it again.



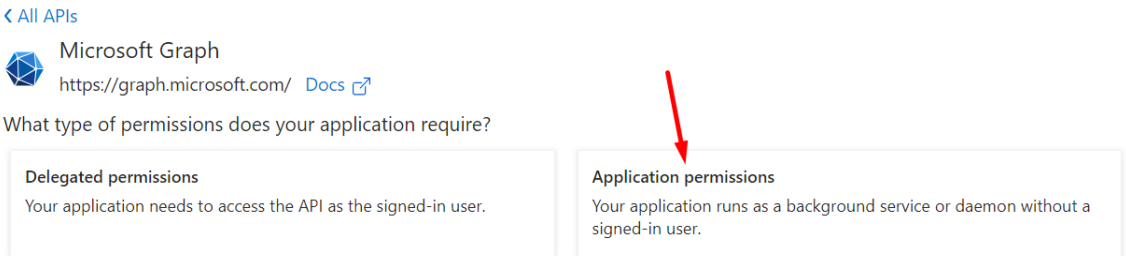
Now we need to give the application some permissions. Go to API permissions and click on Add a permission.



Select Microsoft Graph



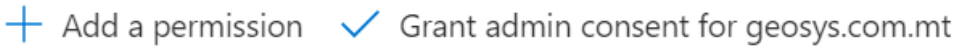
Application permissions



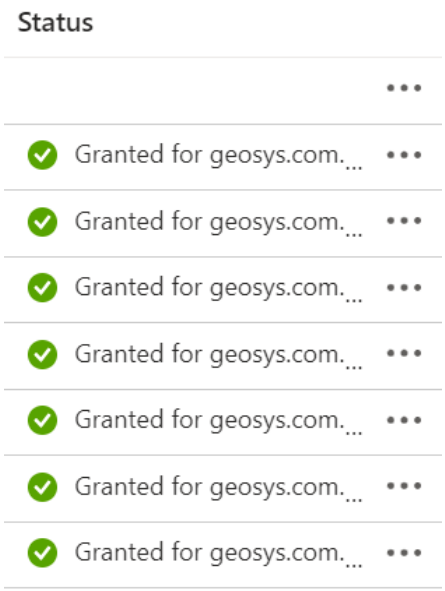
And you will need to give all of the following permissions:

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (7)				•••
Files.ReadWrite.All	Application	Read and write files in all site collections	Yes	Granted for geosys.com... •••
Sites.FullControl.All	Application	Have full control of all site collections	Yes	Granted for geosys.com... •••
Sites.Manage.All	Application	Create, edit, and delete items and lists in all site collec...	Yes	Granted for geosys.com... •••
Sites.Read.All	Application	Read items in all site collections	Yes	Granted for geosys.com... •••
Sites.ReadWrite.All	Application	Read and write items in all site collections	Yes	Granted for geosys.com... •••
TermStore.Read.All	Application	Read all term store data	Yes	Granted for geosys.com... •••
TermStore.ReadWrite.All	Application	Read and write all term store data	Yes	Granted for geosys.com... •••

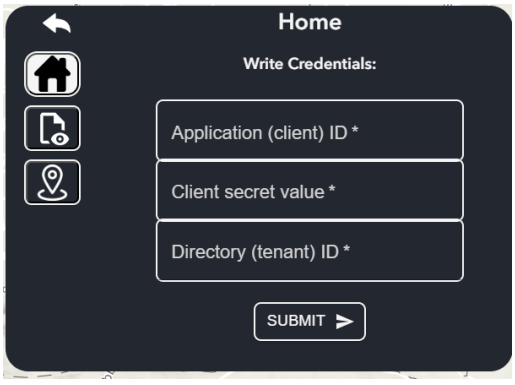
After that, you need to grant admin consent. If you are not a global admin, please ask him to consent it, if not is not going to work.



After that, the status will appear as Granted.

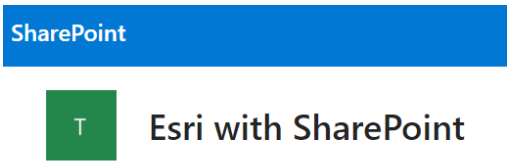


Now with the 3 values that we got, we can now access to the widget

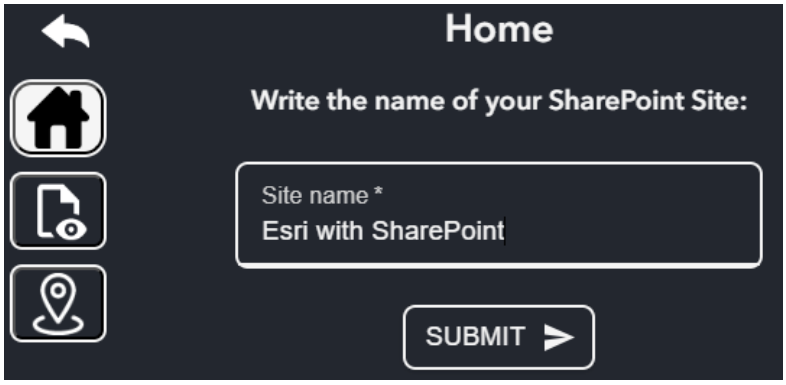


The last step will be to create a SharePoint Column that will store all the tags. After writing the 3 values, you will need to also write the name of the SharePoint site that you want to work with.

As an example, we are going to use this SharePoint site



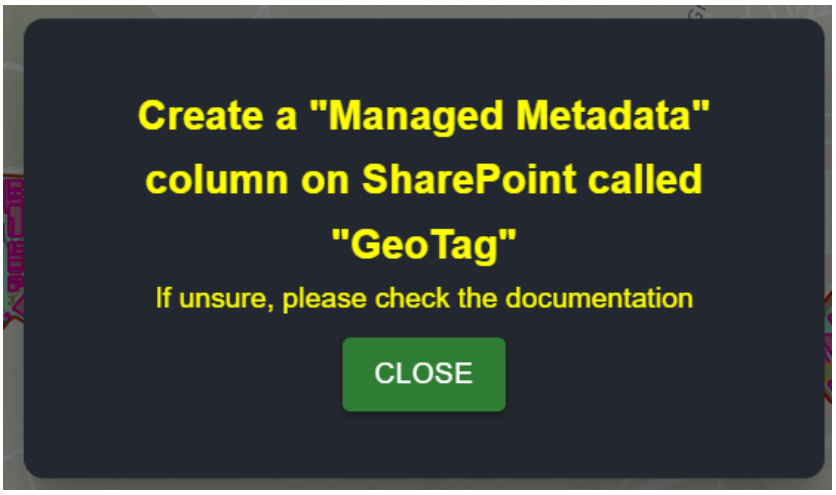
So here we can write the name



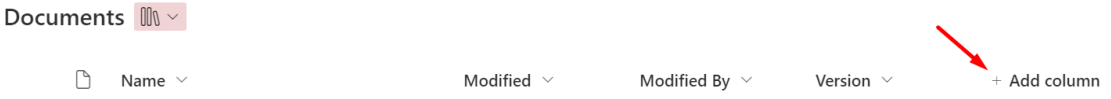
Now please go to “GeoTag” file and choose a random file to work with.

Select the file and on personalized, write a tag for testing. It can be whatever name. Click ADD

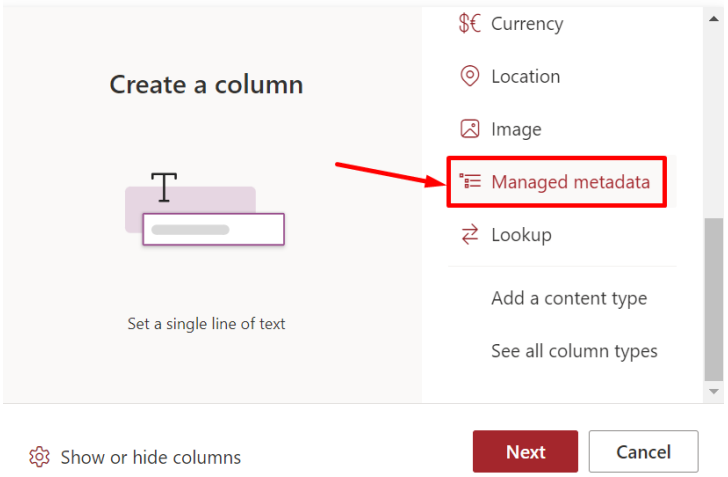
It should appear this warning.



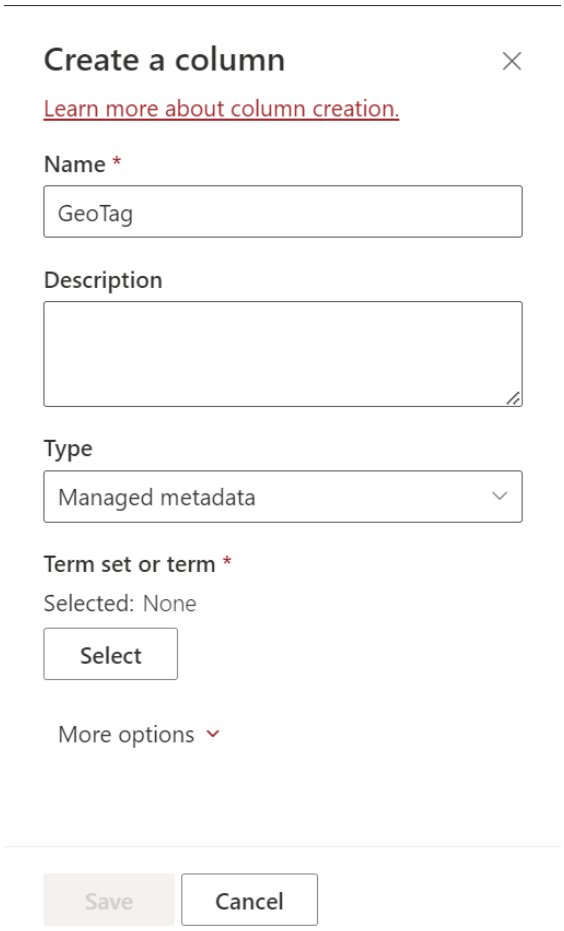
Go to the SharePoint site that you've choosed, then go to Documents and click on Add column.



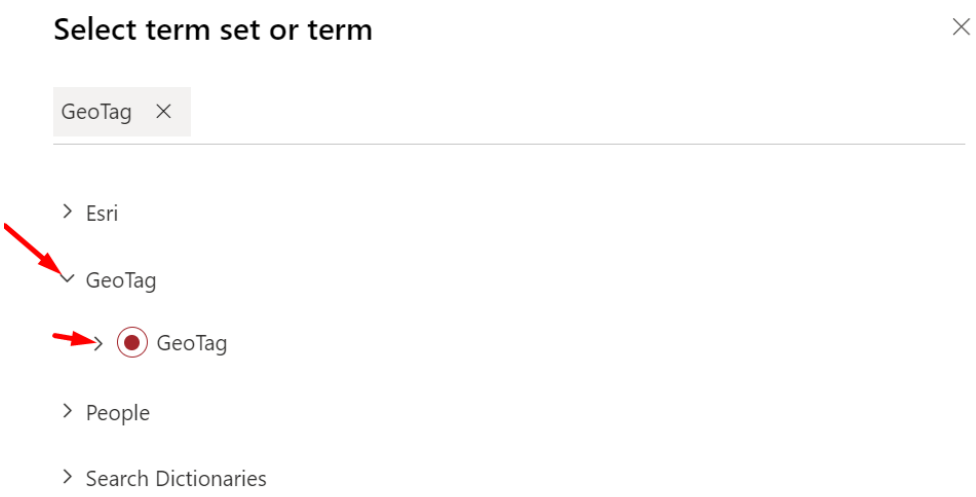
Select "Managed metadata"



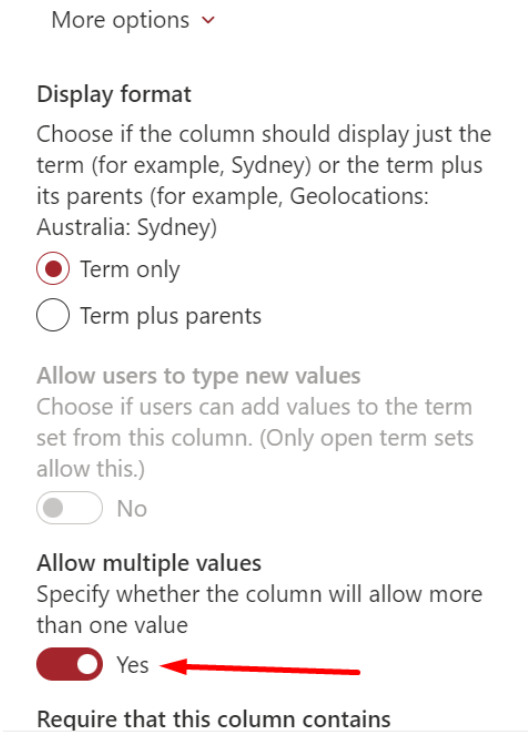
THE COLUMN MUST BE NAMED “GeoTag”. With Upper case G and T. If the name is not identical it will not work.



Then go to Select Term set or term and choose “GeoTag” and then “GeoTag” again:



After that click on More options and then allow multiple values. You don't need to modify the other parameters.



And now it will appear the column GeoTag. Here it will show all the tags that have all the files. If you want you can hide the column, it will still work perfectly.

And that will be it. Please refer to the "GeoTagUsage" Manual for knowing how to work with it.

Bibliography

Github Repositories

GeoTag Widget: https://github.com/martirodm/GeoTag

GeoTag Express: https://github.com/martirodm/GeoTagExpress

Experience Builder Developer Edition

Install guide: https://developers.arcgis.com/experience-builder/guide/install-guide/

ArcGIS Developers: https://developers.arcgis.com/dashboard/

Experience Builder SDK: https://developers.arcgis.com/downloads#arcgis-experience-builder

Microsoft Graph

Official Microsoft documentation: https://learn.microsoft.com/en-us/graph/api/overview?view=graph-rest-1.0

Azure: https://portal.azure.com/

Others

Node: https://nodejs.org/en/download