

Procédure documentation automatique avec node.js et jsdoc

(sur windows10, voir <https://github.com/jsdoc/jsdoc>)

Etape 1 : Comment documenter le code

Les commentaires au format `/**` seront intégrés dans la doc.

```
*  
*/
```

On commente chaque classe et chaque fonction :

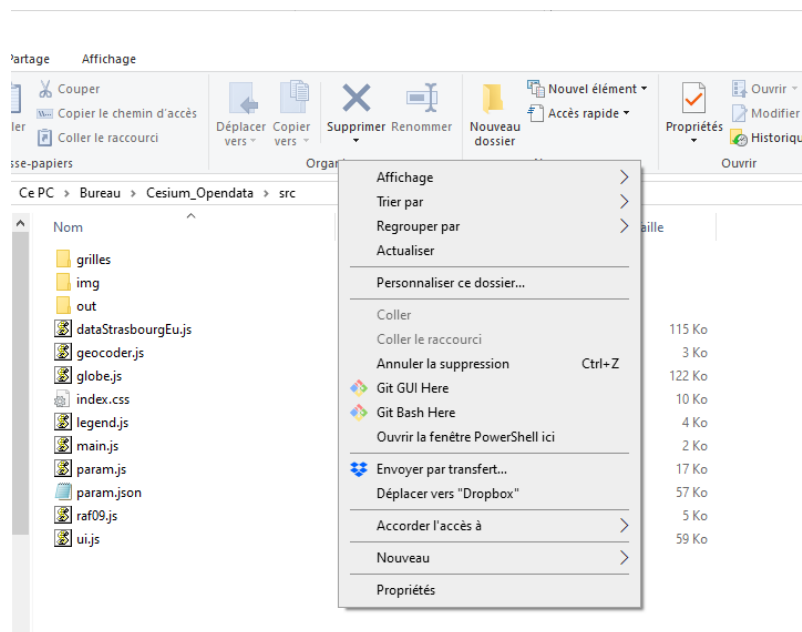
- Explication du rôle de la classe/fonction
- Détails des paramètres avec leur type (string, number, différents objets Cesium), leur nom et leur rôle
- Détail de ce que la fonction renvoie (si elle renvoie qqch)

Voir commentaires existants + ressources internet sur jsdoc

Etape 2 : Création de la documentation avec nodejs

Attention : conserver une copie de l'ancienne doc, particulièrement le index.html qui a été retravaillé (style d'affichage, etc)

- Installer nodejs sur le poste
- Dans le dossier src avec le code faire ctrl + maj + clic droit > ouvrir la fenêtre power shell ici



- Commande `npm install --save-dev jsdoc` (si jsdoc n'est pas encore installé)

- Commande `node_modules/.bin/jsdoc globe.js ui.js param.js legend.js geocoder.js raf09.js tableau.js tile.js` (+ tous les fichiers.js éventuellement rajoutés)

The screenshot shows a Windows PowerShell terminal window with the following text:

```
PS C:\Users\clara\Desktop\Cesium_Opendata\src> ../node_modules/.bin/jsdoc globe.js ui.js param.js legend.js geocoder.js raf09.js
```

- Tous les nouveaux fichiers .js sont à rajouter à la suite de la commande
- Attention au chemin d'accès du fichier jsdoc : si vous avez installé jsdoc dans un autre dossier il faudra redonner le chemin relatif à partir du dossier dans lequel on se trouve (rappel ../ permet de sortir du dossier actuel)
- Un dossier /out est créé ; il contient le index.html (la page d'accueil qui regroupe toutes les classes), chaque documentation de classe, et les fichiers js sous forme d'html (pour être accessibles en ligne dans la doc)

Etape 3 : les actions à faire manuellement

- Le index.html ne doit pas être modifié par les mises à jour avec jsdoc (éléments modifiés et ajoutés comme les onglets Explications et Sources) → penser à copier/coller l'ancien dans le nouveau dossier /out généré
- Le fichier param.json.html doit être mis à jour manuellement en copiant collant le code à la place de l'ancien
- Les .doc concernant la mise à jour de Cesium, la mise à jour de la doc et la génération dynamique du menu avec param.json sont disponibles dans un dossier /ressources du dossier /out → à conserver
- De même pour le doc.drawio qui est le fichier contenant toutes les interactions entre les classes et modifiable sur le site <https://app.diagrams.net/> → une fois les modifications terminées, faire un export doc.png à importer sur le serveur et visible sur la page Image.html

Une fois toutes ces étapes complétées, copier-coller les modifications sur le serveur