

Développement node.js

Partie 1 : classes javascript

La nuit de l'informatique approche. Les étudiants tarnais voulant participer se sont inscrits sur la page web dédié à l'événement. Pour cela, ils ont donné leur nom, leur prénom, leur âge, la formation à laquelle ils appartiennent (BUT MMI, école ISIS, licence Info).

Ces données ont été sous la forme d'un tableau d'objets javascript de type Etudiant. La classe Etudiant a :

- 3 attributs privés nom, prenom, formation
- un constructeur à 3 paramètres,
- 3 getters,
- 1 setter pour la formation,
- une redéfinition de la méthode toString().

Écrire cette classe Etudiant. Utiliser un objet littéral pour représenter les 3 formations (3 attributs, leur valeur étant de type String).

Tester avec le main ci-dessous:

```
const {Etudiant} = require("./Etudiant.js");

const inscrit = new Etudiant("A", "a", "MMI"),

console.log(`étudiant : ${inscrit}`)
```

Écrire cette classe NuitInfo ayant les 3 méthodes suivantes :

- donner le nombre d'étudiants par formation,
- donner la formation la plus représentée,
- lister les étudiants par formation.

Pour la 3^e fonctionnalité, utiliser une collection de type Map. Une collection est une structure de données qui regroupe des éléments de même type et les organisent selon une certaine logique. Javascript propose :

- les ensembles (ou set) https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set
- les dictionnaires (ou map) https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map

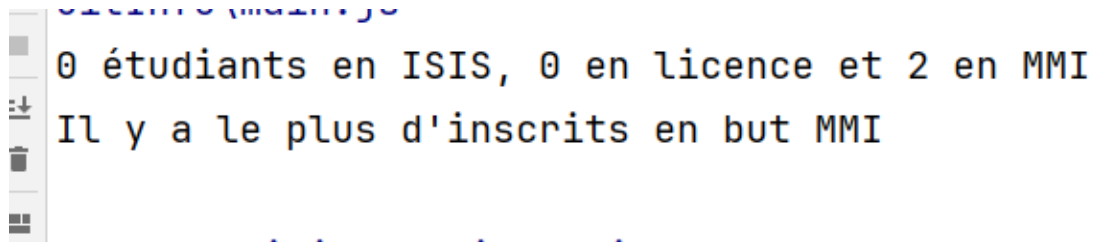
Tester votre code avec le main ci-dessous :

```
const {Etudiant} = require("./Etudiant.js");

const inscrits = [new Etudiant("A", "a", "MMI"), new Etudiant("B", "b", "MMI")]
const formation= { ISIS:"école ISIS", LIC:"licence Info", MMI:"but MMI"}

let [nbISIS, nbLIC, nbMMI]=nbInscritsParFormation(inscrits)
console.log(`${nbISIS} étudiants en ISIS, ${nbLIC} en licence et ${nbMMI} en MMI`)
console.log(`Il y a le plus d'inscrits en ${formationMajoritaire(inscrits)}`)
```

Il doit afficher :



```
0 étudiants en ISIS, 0 en licence et 2 en MMI
Il y a le plus d'inscrits en but MMI
```

Partie 2 : serveur de pages

Écrire une application Express qui sert 2 pages :

- une pour afficher pour chaque formation le nombre d'étudiants inscrits ainsi que leur nom,
- une pour inscrire un nouvel étudiant.

Partie 3 : websockets

Lorsqu'un utilisateur ajoute un nouvel étudiant, les utilisateurs qui sont en train de consulter la liste des étudiants ne sont pas notifiés.

On voudrait que leur page soit mise à jour sans qu'ils aient à refaire une requête. Mettre en place des websockets pour cela.