T1f - Compléments / Affichage des moyennes

Affichage des moyennes

Plutôt que d'afficher les notes des élèves, l'enseignant préférerait afficher leur moyenne.

- Dans la fonction associée à la route `/moyennes`, avant l'instruction `return`, insérez des lignes vides afin d'y écrire le code qui produit le résultat souhaité. Pour cela inspirez-vous des conseils ci-après :
  - Déclarez une variable nommée `moyennes` qui contient un tableau qui est vide au départ.
  - Écrivez une boucle qui parcourt le tableau `eleves`
  - Pour chaque élève :
    - Calculez la moyenne de ses notes en utilisant la fonction `getMoyenne()` qui se trouve dans le module `moyenne.js`
    - Créez un nouvel objet doté des propriétés `prenom` et `moyenne`

      NB. Il n'est pas nécessaire d'utiliser une fonction constructeur pour créer cet objet. Vous pouvez utiliser la syntaxe littérale (avec les accolades).
    - Ajoutez cet objet au tableau `moyennes`
  - À la fin, retournez le tableau `moyennes`

Dans le navigateur, l'affichage en mode réduit doit ressembler à ceci :

```
{ "prenom": "Aline", "moyenne": 13 }
{ "prenom": "Brice", "moyenne": 15 }
{ "prenom": "Cathy", "moyenne": 9 }
{ "prenom": "Diego", "moyenne": 11 }
```

Gestion de plusieurs groupes d'élèves

Création des modules groupe2 et groupe3

- Par copie-coller, dupliquez le fichier `groupe1.js` afin de créer 2 nouveaux fichiers nommés `groupe2.js` et `groupe3.js`
- Modifiez leur contenu afin d'y mettre les listes d'élèves suivantes :

  `groupe2.js`

```
new Eleve('Edith', [9, 9, 12]),
new Eleve('Félix', [10, 9, 5]),
new Eleve('Greta', [16, 14, 12]),
new Eleve('Habib', [14, 15, 19]),
```

Licence Creative Commons BY-SA 4.0 - AMBLARD Emmanuel 2026

groupe3.js

```
new Eleve('Isaac', [14.5, 12.5, 10.5]),
new Eleve('Julie', [13.5, 14.5, 18.5]),
new Eleve('Kader', [12.5, 12.5, 15.5]),
new Eleve('Lydia', [11.5, 10.5, 6.5]),
```

Définition d'une route avec paramètre

- Ouvrez le fichier qui définit la route `/moyennes`
- Modifiez cette route pour qu'elle devienne `/moyennes/:id`
- Dans la fonction, écrivez le code qui permet de récupérer la valeur du paramètre `id` :

  `const id = Number(request.params.id)`

  NB. Un paramètre de route ou d'**URL** est une chaîne de caractères. Il vaut mieux la convertir en nombre lorsqu'elle représente une valeur numérique.
- À la fin de la fonction :
  - Mettez en commentaire : `return moyennes`
  - À la place, écrivez : `return { id }`
- Dans le navigateur, ouvrez l'**URL** avec le paramètre **1** :

  http://[::1]:3000/moyennes/1
- Puis, faites la même chose avec le paramètre 2

  Vous devez constater que la valeur du paramètre est bien récupérée par la fonction.

Contrôle de la valeur du paramètre

- Pour avoir un comportement cohérent, écrivez le code qui renvoie une erreur **404**, si la valeur du paramètre est absente ou n'est pas comprise entre 1 et 3 :

```
if (!id || id < 1 || id > 3) {
  return reply.code(404).send({ message: 'N° de goupre invalide' })
}
```

- Dans le navigateur, essayez une **URL** avec une mauvaise valeur de paramètre.

Importation des modules groupe2.js et groupe3.js

Pour que notre fonction ait accès aux élèves des groupes 2 et 3, il faut les importer depuis les modules qui les contiennent.

- Par copier-coller, au début du fichier, dupliquez l'instruction qui fait l'importation à partir du module **groupe1.js**, afin d'importer également les modules correspondant aux groupes 2 et 3.

Licence Creative Commons BY-SA 4.0 - AMBLARD Emmanuel 2026

Comme on pouvait s'y attendre, l'ajout de ces 2 instructions fait planter le serveur, car la variable `eleves` se trouve définie 3 fois.

En effet, le tableau des élèves porte le même nom dans les 3 modules.

Mais ce n'est pas un problème que les variables d'origine portent toutes le même nom.
Il suffit de leur donner un nom différent au moment où elles sont importées.

- Dans les 3 instructions **import**, faites suivre le nom de la variable `eleves` de la mention `as groupe1` pour le groupe 1, et ainsi de suite pour les autres groupes.

Ainsi, on a 3 variables ayant chacune un nom différent.

Envoi des moyennes du groupe demandé

- Apportez les adaptations nécessaires pour que la fonction retourne un tableau qui contient les moyennes du groupe dont l'identifiant a été indiqué dans la route.
- Ouvrez l'**URL** avec les différents numéros de groupe, pour vérifier que ça fonctionne :
  `http://[::1]:3000/moyennes/1`
  `http://[::1]:3000/moyennes/2`
  `http://[::1]:3000/moyennes/3`

Affichage de la liste complète des élèves

L'enseignant souhaite à présent que la route `/moyennes` (sans paramètre) lui permette d'afficher la liste de tous les élèves, si possible triée sur la moyenne.

Ajout d'une route supplémentaire

Dans le fichier **moyennes.js**, la route `/moyennes` a disparu puisqu'elle a été remplacée par la route `/moyennes/:id`. Vous allez la récréer.

- À l'intérieur du fichier **moyennes.js,** il y a un bloc qui est délimité par l'instruction
  `export default async function (fastify, opts) {`
  et par la toute dernière accolade fermante.
- À l'intérieur de ce bloc, il y a un sous-bloc qui commence à l'instruction
  `fastify.get('/moyennes/:id', async function (request, reply) {`
  et qui se termine, à l'avant-dernière ligne du fichier, avec
  `})`
- Dupliquez ce sous-bloc, et attribuez à sa copie la route `/moyennes` (sans paramètre).

Licence Creative Commons BY-SA 4.0 - AMBLARD Emmanuel 2026

Adaptation du code

- Modifiez le code associé à la route `/moyennes` afin qu'il retourne un tableau contenant tous les élèves avec leur moyenne.
- Avant de retourner ce tableau, faites en sorte qu'il soit trié par ordre croissant des moyennes.
- Dans le navigateur, ouvrez l'**URL** :

  `http://[::1]:3000/moyennes`

Vous devez constater que :

Félix a la moyenne la plus faible qui est de **8**

Habib a la meilleure moyenne, à **16**

Licence Creative Commons BY-SA 4.0 - AMBLARD Emmanuel 2026