

ÉPREUVE SYNTHÈSE VOLET B – 20 %

MISE EN SITUATION ET REMISE

La civilisation de monstres rencontrée dans les explorations aime deux choses les pizzas et le Quidditch. Vous avez réussi avec brio à réaliser les services d'échange pour mettre en place le système permettant d'offrir la livraison de pizza à travers la galaxie. Ils vous demandent par la suite de remettre en fonction les services d'échange pour accéder aux statistiques des équipes et des parties de la Ligue intergalactique de Quidditch.

Les questions 1 et 2 sont associées avec les services d'échange de données, l'application serveur qui est située dans le dossier `backend`. N'oubliez pas de faire la commande `npm i` à l'intérieur de ce dossier¹. Vous retrouvez à la racine du dossier `backend` un fichier `tests.json` à importer dans Postman pour vous permettre de tester chacune des sous-questions. Vous devez utiliser `npm start` pour démarrer votre serveur.

La question 3 est associée avec l'application test qui est située dans le dossier `frontend`. Vous n'avez qu'à démarrer le `liveserver` sur le fichier `index.html` pour commencer vos tests.

Cet examen compte 20% de la note finale du cours et doit être réalisé individuellement. Ne pas supprimer pas les TODO présents dans les fichiers.

REMISE

Vous devez remettre dans Léa à la fin de votre examen l'ensemble des fichiers excluant le dossier `node_modules` dans un dossier compressé zip nommé de la première lettre de votre prénom suivi de votre nom et de ESB (ex : YCharronESB).

QUESTION 1 - ÉQUIPES

7 POINTS

Les cinq prochaines sous-questions sont en lien avec la ressource `Team`.

QUESTION 1A – PARAMÈTRE D'URL

1,5 POINTS

Ajouter le code à l'objet `options` qui permet de récupérer l'identifiant de l'équipe demandée et qui est contenu dans l'URL de la route `getOne`.

Le code à modifier débute à la ligne 16 du fichier `routes/teamsRoutes.js`

QUESTION 1B – RETOURNER UNE RÉPONSE

0,5 POINT

Faites-en sorte que la route `getOne` retourne une réponse au client avec le statut HTTP OK et l'équipe possédant le paramètre d'URL fourni.

Le code à développer est à la ligne 29 du fichier `routes/teamsRoutes.js`

QUESTION 1C – ERREUR INTERNE

0,5 POINT

Ajouter le code qui permet de traiter les erreurs pouvant survenir dans le bloc `catch` de la route `getOne`.

Le code à développer débute à la ligne 32 du fichier `routes/teamsRoutes.js`

¹ `cd backend`

QUESTION 1D -TRANSFORMATION

2 POINTS

Faites-en sorte que les équipes soient transformées avant d'être envoyées au client. Voici les transformations à faire dans la méthode `transform` de la classe `TeamsService`.

- Ajouter un `href` à l'équipe
- Ajouter un `rating` à l'équipe correspondant à la moyenne des `ratings` des joueurs de l'équipe, attention la moyenne doit être un nombre et avoir 2 chiffres après la virgule.

```
"href": "http://localhost:8526/teams/5fd255995243436794040b3d",  
"rating": 89.04
```

Le code à développer débute à ligne 22 du fichier `services/teamsService.js`

QUESTION 1E – PARAMÈTRE PLAYER

2,5 POINTS

Ajouter la possibilité pour les clients du service d'échange d'utiliser le paramètre d'URL `player` dans la route `getAll` de la ressource `Team`. Lorsque le paramètre est fourni, la route doit retourner seulement les équipes dont un de leurs joueurs est nommé avec la valeur du paramètre. Vous devez ajouter le code pour traiter la réception du paramètre dans la route `getAll` et modifier la méthode `retrieveAll` du service pour utiliser le paramètre.

Le code à développer est à la ligne 40 du fichier `routes/teamsRoutes.js` et à la ligne 13 du fichier `services/teamsService.js`.

QUESTION 2 - PARTIES

6 POINTS

Les trois prochaines sous-questions sont en lien avec la ressource `Game`

QUESTION 2A – PAGINATION

2,5 POINTS

1. Ajouter dans les paramètres nécessaires aux méthodes `limit`, `skip` et `sort` pour activer la pagination triée par ordre décroissant de date de partie (`gameDate`);
2. Développer la requête Mongoose pour compter le nombre de parties existantes en base de données;
3. Modifier le retour de la méthode pour permettre à la route de fonctionner adéquatement.

Le code à développer débute à la ligne 12 du fichier `services/gamesService.js`

QUESTION 2B - TRANSFORMATION

1,5 POINTS

Faites-en sorte que les parties soient transformées avant d'être envoyées au client. Les propriétés virtuelles doivent être retournées dans la réponse.

Le code à modifier est à la ligne 33 du fichier `routes/gamesRoutes.js`.

QUESTION 2C– RÉSULTAT DE LA PARTIE

2 POINTS

Ajouter à la transformation d'une partie la propriété `result` indiquant le résultat du match. Vous devez utiliser l'objet `score` de la partie.

3 scénarios sont possibles :

```
"score": {
  "home": 380,
  "away": 260
}
```

- L'équipe à domicile (`home`) a un nombre de point supérieur à l'équipe visiteuse (`away`) et l'emporte;
- L'équipe visiteuse (`away`) a un nombre de points supérieur à l'équipe à domicile (`home`) et l'emporte;

```
"result": "Ucarvis l'emporte 380 à 260 contre Pulreshan"
```

- Les deux équipes ont le même nombre de points et la partie est nulle.

```
"result": "Partie nulle - 350 à 350 entre Postraigantu et Thaepra"
```

Le code à développer est à la ligne 31 dans la méthode `transform` du fichier `services/gamesService.js`

QUESTION 3 - APPLICATION TEST

7 POINTS

Les quatre prochaines sous-questions sont en lien avec l'application test située dans le dossier `frontend`.

QUESTION 3A – LES ÉQUIPES

2 POINTS

La première étape est d'ajouter l'ensemble des équipes retourner par la requête Ajax dans le tableau `#teams` de la page `index.html`. Vous devez utiliser la fonction `generateTeamHTML`.

Le code à développer se retrouve à ligne 18 dans la fonction `getTeams` du fichier `js/index.js`

Résultat attendu après la question 3A

Équipes du championnat de Quidditch

Planet	Entraîneur	Joueurs	Évaluation
Stretelara			94.86
Inzinda			90.17
Kucruigawa			91.48
Becipra			91.53
Phiuq SIV			91.6
Griuq 6ZML			92.06
..			---

QUESTION 3B - L'ENTRAINEUR

1 POINT

Modifier les attributs `src` et `title` de la balise `img` pour afficher l'image et l'infobulle de l'entraîneur dans la 2^e colonne du tableau. L'infobulle doit afficher : Coach – [nom de l'entraîneur]. Il faut remplacer les `[]` par le nom de l'entraîneur.



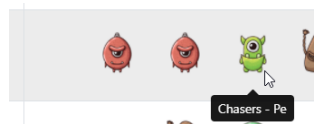
La propriété `title` permet d'afficher l'infobulle quand le curseur passe sur l'image, le code est déjà fonctionnel pour cela, vous devez seulement modifier l'attribut `title`.

Le code à modifier se retrouve à la ligne 32 dans la fonction `generateTeamHtml` du fichier `js/index.js`

QUESTION 3C – L'ALIGNEMENT

2 POINTS

1. Ajoutez le code permettant d'afficher l'image de chacun des joueurs de l'équipe dans la colonne 3^e colonne du tableau.
2. Modifier les attributs `src` et `title` de la balise `img` pour afficher l'image et l'infobulle de chacun des joueurs d'une équipe. L'infobulle doit afficher : [position du joueur] – [nom du joueur]. Il faut remplacer les `[]` par la position du joueur et son nom.



La propriété `title` permet d'afficher l'infobulle quand le curseur passe sur l'image, le code est déjà fonctionnel pour cela, vous devez seulement modifier l'attribut `title`.

Le code à développer débute à la ligne 36 dans la fonction `generateTeamHtml` et à la ligne 48 dans la fonction `generatePlayerHtml` du fichier `/js/index.js`

QUESTION 3D – L'ÉVALUATION

2 POINTS

Faites en sorte de faire de la mise en forme sur l'évaluation (`rating`) de l'équipe, si l'évaluation est supérieure ou égale à 90 vous devez mettre le texte en vert, si elle est inférieure le texte doit être affiché en rouge. Deux classes css `green` et `red` sont déjà présentes dans le fichier `/css/style.css`.

Résultat attendu après la question 3D

Porornania			88.93
Neon 5			90.39
Ucarvis			90.03
Phapitis			89.48

90.52

88.55

Fin de l'épreuve synthèse Volet B – bon repos à vous tous, on se retrouve en janvier