420-4W6 - TP3 - Flappy Birb (15%)

# Consignes

* Le projet sera fait de façon individuelle.
* Le projet client sera fait avec le Framework Angular. (Un projet de départ est fourni !)
* Le projet serveur sera fait avec le Framework .Net Core.

# Mise en contexte

Sur l’application client, un jeu est déjà fonctionnel. Vous contrôlez un oiseau (**W** pour monter, **E** pour changer de couleur) et il faut franchir le plus d’obstacles possible sans les toucher. Notez que l’oiseau, qui peut alterner entre le vert et le bleu avec la touche **E**, peut toucher les obstacles de la même couleur que lui sans problème.

# Gestion des utilisateurs (Semaine 9)

Vous devrez permettre l’inscription d’un nouvel utilisateur, la connexion d’un utilisateur existant et la déconnexion d’un utilisateur.

# Créer un score (Semaine 9)

Après avoir terminé une partie, on peut poster son score au serveur si on est connecté.

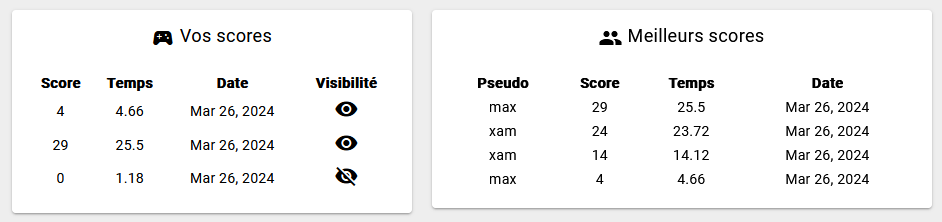
# Afficher nos scores (Semaine 9)

Un utilisateur connecté doit pouvoir consulter la liste de ses scores.

# Afficher tous les scores publics (Semaine 10)

On doit pouvoir consulter la liste des 10 meilleurs scores publics, qu’on soit connecté ou non.

Exemple si je suis connecté avec un utilisateur nommé **max** :



# Modifier la visibilité de nos scores (Semaine 10)

Un utilisateur connecté doit pouvoir changer la visibilité (public ou privé) de ses scores lorsqu’il les consulte. Cela va bien entendu déterminer si les autres utilisateurs peuvent apercevoir les scores dans la liste de tous les scores publics. La visibilité doit être la SEULE propriété modifiable.

# Seed de données (Semaine 10)

À des fins de tests, la base de données doit initialement contenir deux utilisateurs et quatre scores : c’est-à-dire deux utilisateurs avec chacun un score public et un score privé.

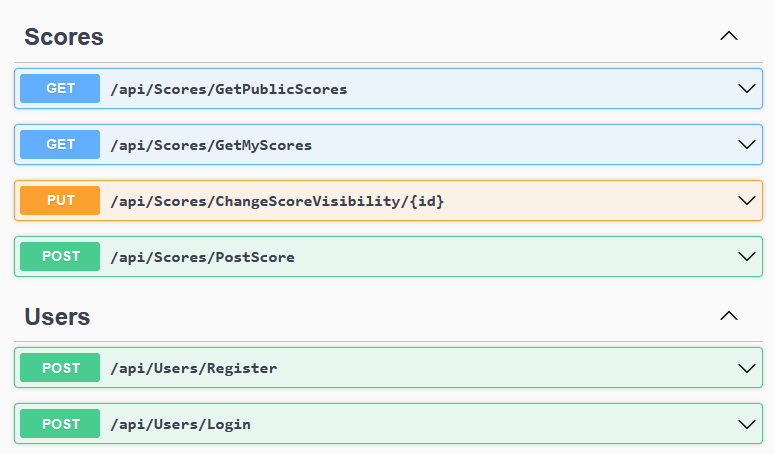
# Suppléments agaçants

* L’accès aux données et la modification des données doit être **sécuritaire** (Semaine 9 et 10)
* Le projet ASP.NET Core doit posséder un **service** pour les scores. (Semaine 8)
* Le projet Angular doit utiliser un **service** pour les requêtes HTTP. (Semaine 3)
* Un **intercepteur** doit être utilisé pour joindre le token d’authentification aux requêtes dans le projet Angular. (Semaine 10)

# Correction

|  |  |  |
| --- | --- | --- |
| 6.5 pts | Gestion des scores | |
|  | 2 pts | Créer un score |
| 2 pts | Changer la visibilité d’un score (privé / public) |
| 1.5 pts | Afficher les scores publics |
| 1 pt | Afficher les scores de l’utilisateur connecté |
| 3.5 pts | Gestion des utilisateurs | |
|  | 1 pt | Inscription |
| 1.5 pts | Connexion |
| 1 pt | Déconnexion |
| 5 pts | Divers | |
|  | 1.5 pts | Sécurité (On ne peut pas obtenir les données d’autres utilisateurs, voir les scores privés des autres, modifier la visibilité des scores des autres ou modifier autre chose que la visibilité d’un score) |
| 1 pt | Services côté serveur (Couche entre les contrôleurs et le DbContext) |
| 0.5 pt | Utiliser un interceptor (Angular) |
| 1 pt | Données de test dans la BD (Seed) |
| 0.25 pt | Architecture Angular bien organisée (Usage de services, de modèles, de composants) |
| 0.75 pt | Utilisation raisonnable de Git (Inviter l’enseignant, commits clairs, respect des conventions du département et au moins 10 commits cohérents) |
| 15 pts | Total | |

Liste des requêtes disponibles :



⛔ Précision sur un bug mineur connu :

Dans le projet Angular de départ, lorsqu’on quitte le composant Play, la page Web est réactualisée. (C’est nécessaire pour que l’état du jeu ne se duplique pas en revenant plus tard sur Play) Donc si on est sur le composant Play et qu’on va vers un autre composant (Login ou Scores), on observera un petit hoquet et la page Web sera rafraîchit.

Un bug courant est d’observer une erreur dans la console du navigateur lorsqu’on lance une requête dans la fonction ngOnInit() du composant Scores. Pas de panique, la requête restera fonctionnelle une fois la page rafraîchit et vous ne perdrez pas de points. Mettre une requête à cet endroit risque d’être nécessaire dans tous les cas.