



PROJET OIL-2022

SCRABBLE-TD3-WINNERS

Les Participants :

- ✓ *DIALLO Fatoumata Binta*
- ✓ *DIEYE Khadidiatou*
- ✓ *OUAZRI Safouane*
- ✓ *OUSSAMA Samia*
- ✓ *BOUSADIA Lahcene*
- ✓ *MAHDI Chaymae*

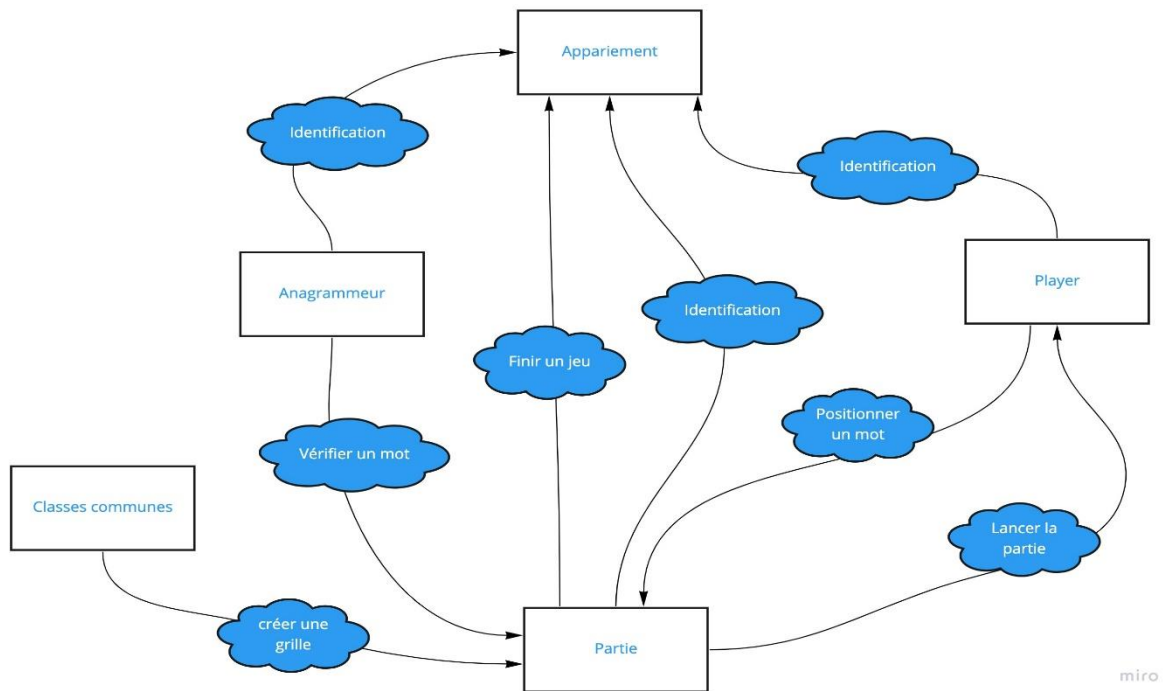
Professeur :

Monsieur Philippe Renvier Gonin

Année scolaire : 2021 – 2022

Réponses aux questions

1. Faites un diagramme général de l'architecture de vos services, en expliquant le rôle de chaque service



2. Détaillez les échanges entre les services (la forme est libre)

Dans le projet nous avons 5 composants et 8 webservices.

Les 5 composants sont : L'anagrammeur, l'appariement, la partie, le player et les classes communes.

- L'anagrammeur qui gère un dictionnaire de mot ;
- La partie qui est le jeu, là où les joueurs jouent ;
- Le player dans notre application qui n'est pas un humain, mais joue le scrabble ;
- Les classes communes qui comportent l'ensemble des classes qui sont utilisés par plusieurs composants.
- L'appariement qui permet de réunir tous les autres composants et les connectés entre eux ;

Ces composants ci-dessus, s'échangent grâce à des services. Ces derniers sont :

- Trois services d'identification permettant aux composants partie, player et anagrammeur de se connecter à l'appariement.
- Un service 'créer une grille' pour permettre la création de la grille. L'échange s'effectue entre la classe commune et la partie. La grille est créée dans la classe commune, et une api « /grid » est créée dans la classe PartieWebController. ET grâce à cet API que l'échange s'effectue entre la partie et la classe grid.
- Un service 'lancer la partie' qui permet de lancer une partie. Pour qu'une partie soit lancée, il faut deux joueurs. ET une deux joueurs dans le plateau, la partie est lancée automatiquement.
- Un service « positionner un mot » qui permet de positionner un mot dans la grille. L'échange s'effectue entre player et partie.
- Un service « Vérifier un mot » qui reçoit un mot et envoie à la partie si le mot existe ou non.
- Un service « finir un jeu » qui permet de finir une partie. L'échange s'effectue en l'appariement et partie.

3. Pour chacun de vos services précisez s'ils sont stateless ou non et expliquez pourquoi.

Un service est dit "stateless" s'il traite la demande uniquement sur la base des informations transmises dans chaque demande et ne s'appuie pas sur les informations de la demande précédente. Autrement dit, le serveur n'a pas besoin de conserver les informations d'état entre les requêtes (ou l'état peut être stocké dans un service externe tel que : serveur). Base de données).

Dans notre cas :

- Création de la grille : Service Stateless car il ne dépend pas sur des informations d'une demande précédente.

- Lancer la partie : Service non Stateless car pn peut pas lancer la partie si on n'a pas encore créé une grille.
- Positionner un mot : Service non Stateless, il dépend des informations du lancement du partie.
- Vérifier un mot : Service non Stateless, il dépend des informations du positionnement du mot.
- Finir un jeu : là on peut dire que le Service Stateless car il dépend des informations des demandes précédentes comme le lancement du partie, création du grille, positionnement des mots...

4. Expliquez vos tests en les regroupant par types de tests (unitaires, unitaires dans spring, intégration, etc.) et en une phrase par fichier de tests expliquez ce qui est testé.

Nous avons effectué un seul test qui été de type Unitaire, ce test a pour but de valider la connexion entre la partie et les autres services.