

ATELIER II -16h-

COMPETENCES

Concepts :

- Différentes architectures de conception logicielle
 - SOA
 - Bus de communication
 - Websocket
 - Proxy
- Communication multi-techno en BUS
- Websocket
- Redirection de requête
- Interaction Back-Front

Techno :

- ActiveMq
- NodeJS, socket.io
- React, Javascript, AJAX

SUJET

1. PHASE A :

L'entreprise souhaite maintenant mettre en place une interface de jeux et un chat entre les utilisateurs. Etant référant sur le projet vous justifiez de la mise en place d'un autre backend basé sur NodeJS.

- 1.1. Expliquer les avantages et inconvénients de Node.js pour ce type d'activité.
- 1.2. Réaliser un backend Node.js indépendant de l'existant permettant de créer un chat entre deux utilisateurs (javascript).

FAIRE VALIDER VOTRE PRODUCTION

- 1.3. Mettre à jour votre application afin de pouvoir récupérer la liste des utilisateurs via Web service sur l'application précédemment créée

FAIRE VALIDER VOTRE PRODUCTION

2. PHASE B :

L'entreprise ne dispose pas de FrontEnd et souhaite utiliser un Framework FrontEnd pour le réaliser

- 2.1. Réaliser en BackEnd Node.js un service permettant d'associer les utilisateurs souhaitant jouer entre eux.

FAIRE VALIDER VOTRE PRODUCTION

- 2.2. Réaliser en un diagramme de séquence expliquant les interactions entre Node.js et le FrontEnd lors d'un jeu entre deux joueurs
Règles : chaque joueur a sélectionné X cartes

Le maitre du jeu décide qui commence.

Chaque utilisateur reçoit des points d'actions utilisés pour jouer des cartes.

Chaque utilisateur a l'ensemble des cartes visibles sur le plateau.

L'utilisateur qui commence (u1), sélectionne une de ces cartes puis sélectionne une carte adverse et décide d'attaquer (des points d'action sont dépensés).

Le maitre du jeu calcul de résultat de l'attaque (ajout d'aléatoire, comme critique, esquive etc..)

Une fois que u1 n'a plus de points d'action ou clic sur le bouton « enturn », la main passe au joueur 2. Les points d'actions non joués se cumulent.

La partie se termine lorsque qu'un joueur n'a plus de carte.

Le maitre du jeu décide de la récompense à attribuer (argent).

2.3. Réaliser le BackEnd en Node.js et le FrontEnf en React.

3. PHASE C :

3.1. Gérer la sauvegarde de l'historique des conversations du Chat en envoyant les données du BackEnd Node.js vers le BackEnd Springboot qui stockera ces informations.

3.2. Informer le BackEnd SpringBoot gérant les utilisateurs et les cartes du résultats du jeu par Bus de communication (mise à jour de somme de chaque joueur).

FAIRE VALIDER VOTRE PRODUCTION

4. PHASE D :

4.1. Mettre en place un proxy permettant de rediriger les requêtes http entre les 2 backends : Springboot et Node.js

FAIRE VALIDER VOTRE PRODUCTION

4.2. Merger l'ensemble de votre application front.

FAIRE VALIDER VOTRE PRODUCTION

TIPS

- Afin de paralléliser le travail, autoriser le **CROSS ORIGIN** sur vos navigateurs web **ET** sur votre backend
- La réponse du node.js sur le Bus de communication doit être un String du format suivant :
`{"gameId":15,"user1Id":6,"user2Id":12,"moneyOperation1":100,"moneyOperation2":-100}`

BONUS

- Créer un BackEnd indépendant (Springboot) permettant de logger dans un fichier l'ensemble des messages transmis sur le bus de communication.
- Mettez en place un loadbalancer permettant de répartir les joueurs sur différents serveur Nodejs.

QUESTIONS

Qu'est ce que les websockets ? Comment fonctionne-t-elle ?

Quelle est la différence entre socket.io et les web sockets ?

Qu'est ce que npm ? comment fonctionne-t-il ?

Qu'est ce qu'une fonction de callback ?

Expliquer la différence entre une programmation séquentielle et une programmation événementielle.

Pourquoi a-t-on besoin d'utiliser un proxy pour qu'un web browser puisse communiquer avec deux backend différents ?

Quels sont les principaux avantages de Node.js vis-à-vis d'autre serveur web (E.g apache tomcat + Springboot) ?

La communication via websocket consomme-t-elle plus de ressource côté serveur ?

Comment peut-on réaliser une authentification pour une application qui possède des backends distincts ?

Quels arguments/situations peuvent conduire à l'usage d'un bus de communication pour une communication entre backends plutôt que les webServices ?

Qu'est ce que la CI ? expliquer ses avantages.

Qu'est ce que la CD ? Que permet-elle de faire ?

Pourquoi docker devient-il un outil très utile sur des architectures multi-backend et pour la CD ?

Qu'est-ce qu'un callback ? Qu'est-ce qu'une promise ? Qu'est-ce qu'un observable ?

Alternative à activemq ?

Qu'est-ce que Spring Framework par rapport à SpringBoot ?

Quelle partie de SpringBoot permet de créer des web services REST ?

Qu'est-ce qu'ExpressJS ? Quels sont les alternatives ?

Quel est le transport utilisé par Spring pour communiquer avec activemq ?

Quel est le transport utilisé par Node.js pour communiquer avec activemq ?

REFERENCES

***(*indispensable*) **(*important*) *(*optionnel*)

[NGINX]

- *** <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>

[Nodejs]

- ** <https://nodejs.org/en/>
- ** <https://nodejs.org/en/docs/guides/>
- ** <https://github.com/gdaws/node-stomp>
- ** <https://expressjs.com/fr/4x/api.html>
- ** <https://socket.io/docs/>

[Ci-CD]

- ** <https://docs.gitlab.com/ee/ci/>

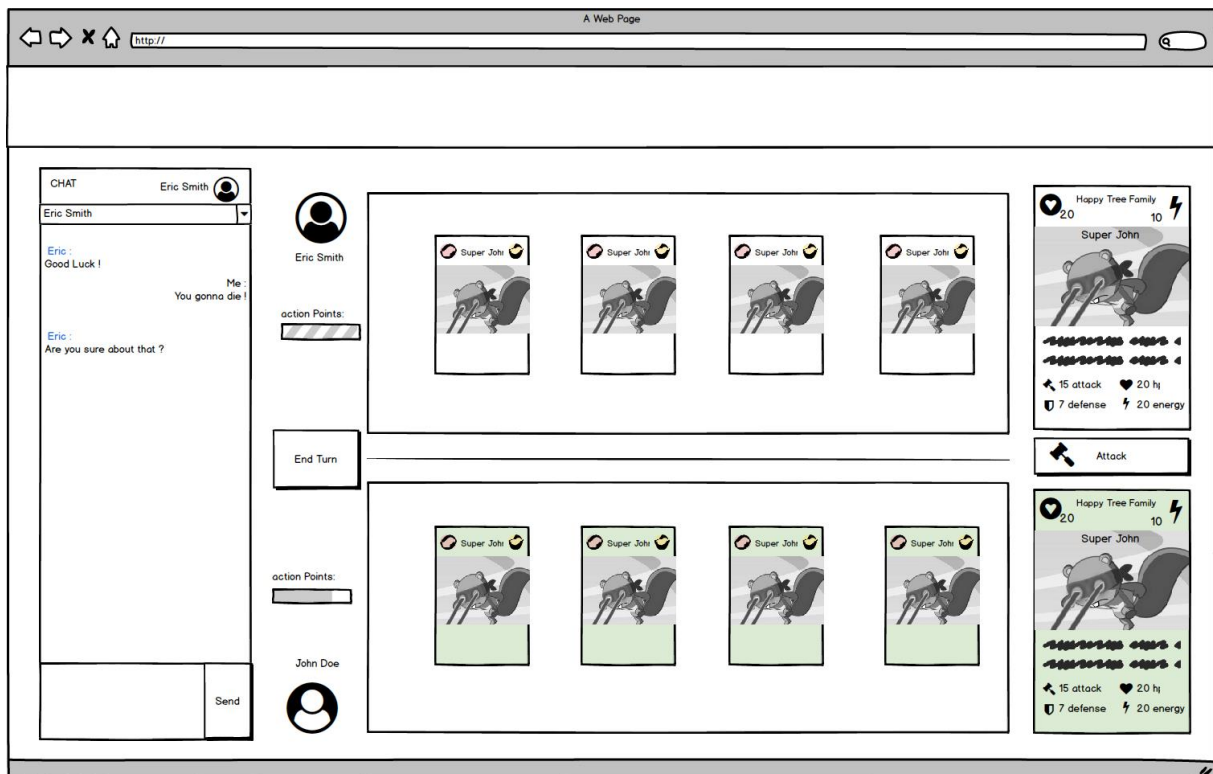
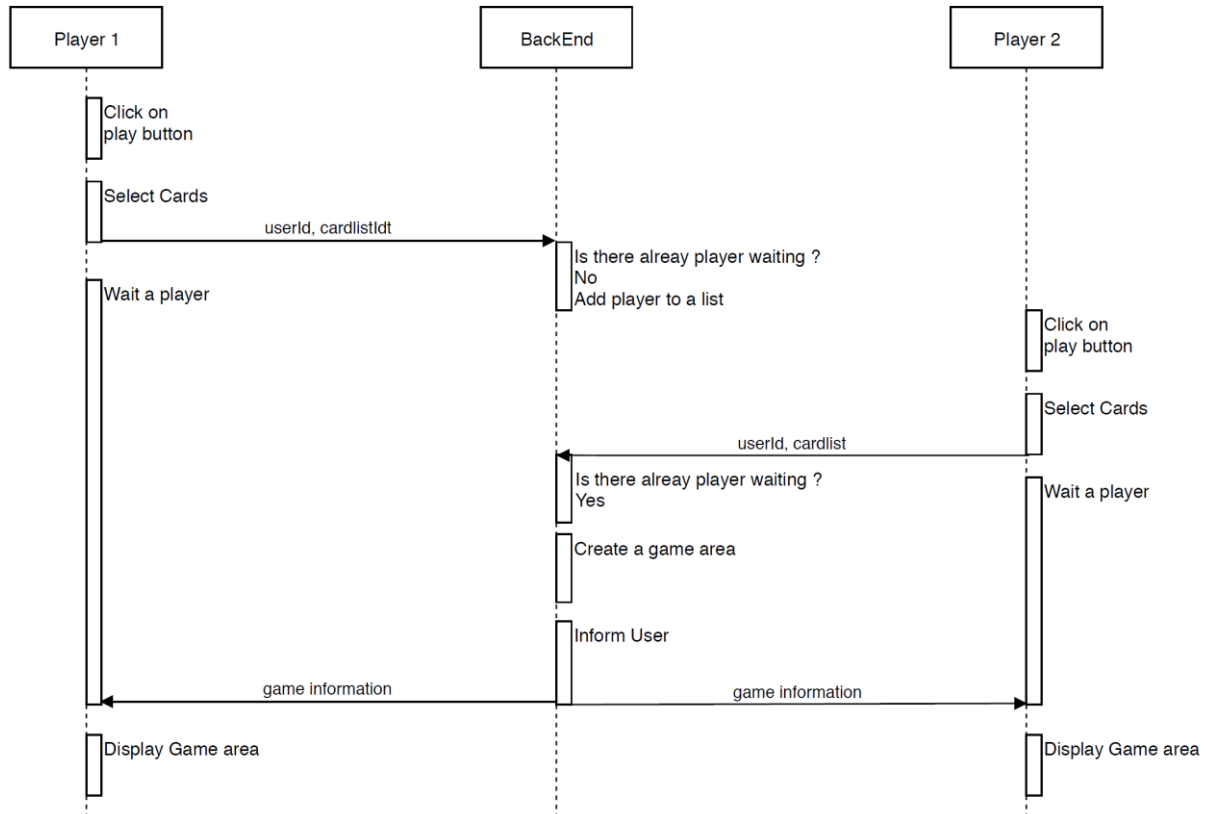
[Backend+BUS]


- ** <https://github.com/gdaws/node-stomp>
- ** <https://spring.io/guides/gs/messaging-jms/>

ELEMENTS DONNES

- Workflow de sélection de joueur
- Spécification d'écrans de l'air de jeu
- Exemple Html Css du chat, carte aire de jeu

Game Area Creation workflow







Happy Tree Family


20

10




Super John







15 attack




20 hp



7 defense



20 energy

CHAT		Eric Smith	
Eric Smith			▼
<p>Eric : Good Luck !</p> <p>Me You gonna die</p> <p>Eric : Are you sure about that ?</p>			
		Send	

Eric Smith	
Eric Smith	
John Doe	
Kity Poulet	
Chuck Norris	

Eric Smith

Eric :
Good Luck !

Me :
You gonna die !