

# ATELIER II -16h-

## COMPETENCES

Concepts :

- Différentes architectures de conception logicielle
  - SOA
  - Bus de communication
  - Websocket
  - Proxy
- Communication multi-techno en BUS
- Websocket
- Redirection de requêtes
- 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 Node.js.

- 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 souhaite permettre aux joueurs de s'affronter lors de matchs

- 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 un diagramme de séquences expliquant les interactions entre Node.js et le FrontEnd lors d'un jeu entre deux joueurs

Règles :

1. Chaque joueur a sélectionné X cartes
2. Le maitre du jeu décide qui commence.
3. Chaque utilisateur reçoit des points d'actions utilisés pour jouer des cartes.
4. Chaque utilisateur a l'ensemble des cartes visibles sur le plateau.
5. 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).
6. Le maitre du jeu calcul le résultat de l'attaque (ajout d'aléatoire, notion de coup critique, esquive etc..)
7. Une fois que  $u1$  n'a plus de points d'actions ou clic sur le bouton « *endturn* », la main passe au joueur 2. Les points d'actions non joués se cumulent pour les tours suivants.
8. 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 FrontEnd en React.

**FAIRE VALIDER VOTRE PRODUCTION**

### 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 résultat du jeu par Bus de communication (mise à jour de somme de chaque joueur).

**FAIRE VALIDER VOTRE PRODUCTION**

### 4. PHASE D :

- 4.1. Créer un BackEnd indépendant (Springboot) permettant de logger l'ensemble des messages transmis sur le bus de communication.

### 5. PHASE E :

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

**FAIRE VALIDER VOTRE PRODUCTION**

- 5.2. Merger l'ensemble de votre application front.

**FAIRE VALIDER VOTRE PRODUCTION**

### 6. PHASE F (option):

- 6.1. Containeriser vos différents backend (Dockerfile)
- 6.2. Déployer vos services à l'aide de dockercompose

**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,"userId":6,"userId":12,"moneyOperation1":100,"moneyOperation2":-100}`

### **BONUS**

- Si vous avez réalisé l'étape F, mettre en place du déploiement continu.
- Mettez en place un loadbalancer permettant de répartir les joueurs sur différents serveurs 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/>

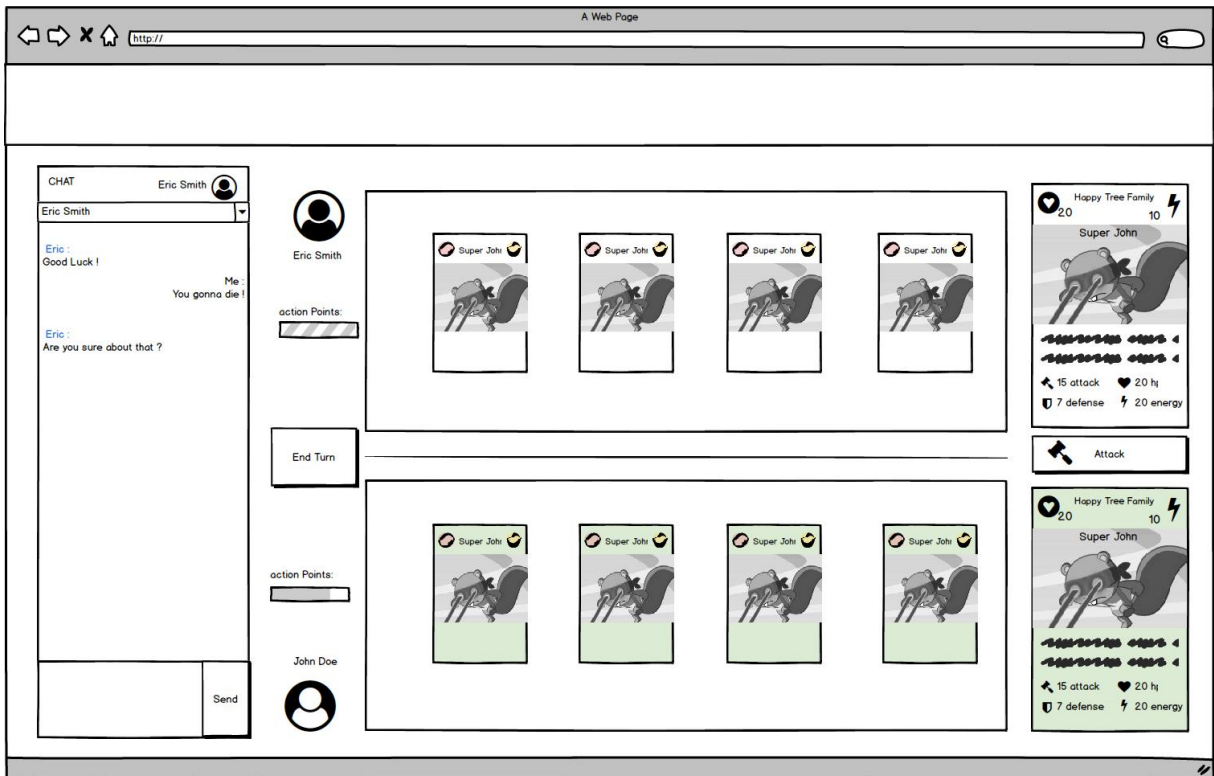
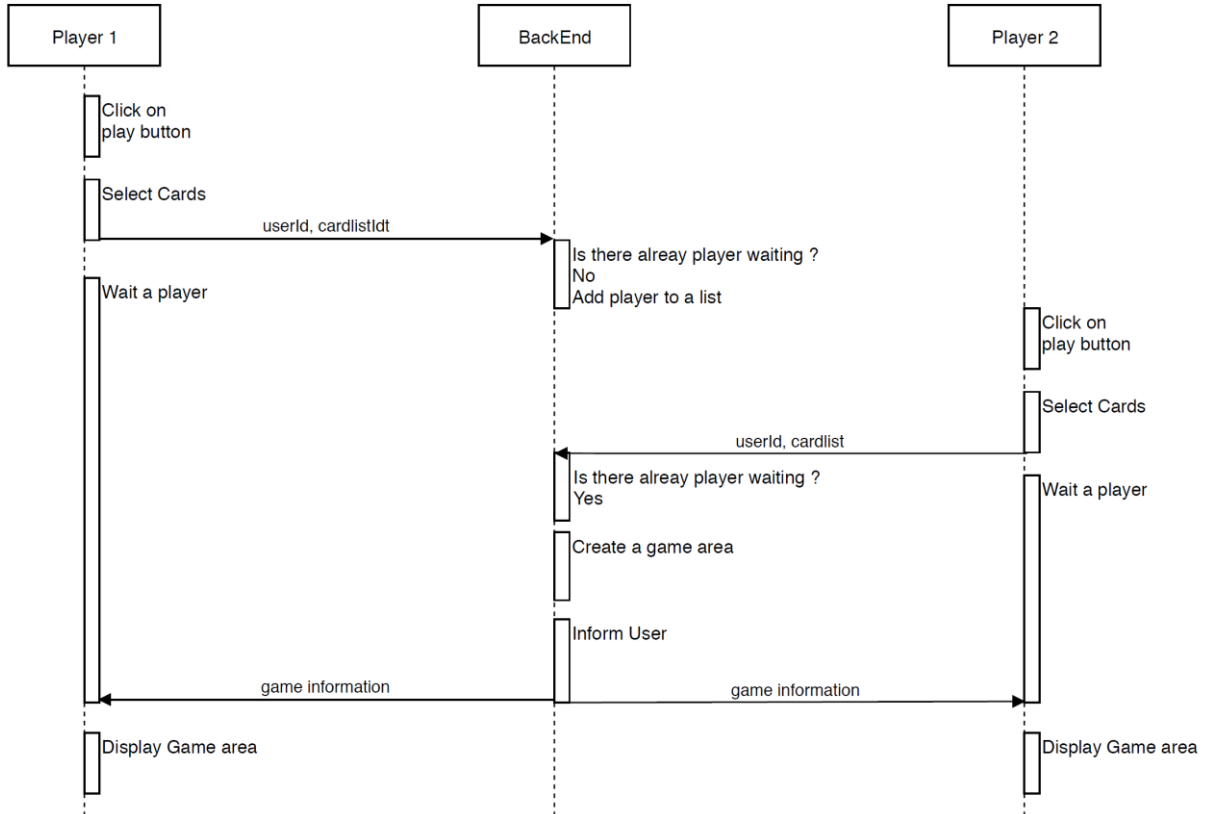
### [Backend+BUS]

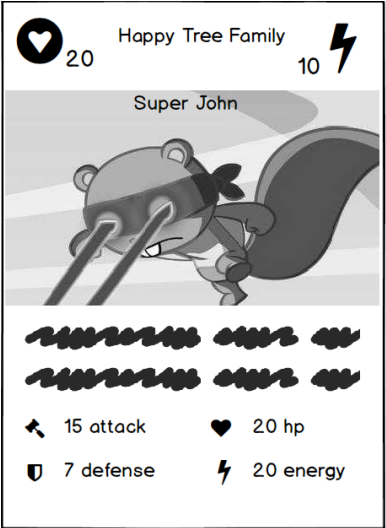
- \*\* <https://github.com/gdaws/node-stomp>
- \*\* <https://spring.io/guides/gs/messaging-jms/>

## 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





Full card  
View



Short Card  
View



Eric Smith

Eric Smith

John Doe  
Kity Poulet  
Chuck Norris

Combobox  
list of users

Eric Smith

Current  
selected  
user and  
associated  
name

Eric :  
Good Luck !

Me :  
You gonna die !

Chat historic  
exchanges