

ATELIER I -12h-

COMPETENCES

Concepts :

- Différentes architectures de conception logicielle
 - MVC
 - SOA
 - Bus de communication
- Avantages/inconvénients des architectures
- Monolithique to SOA
- Découpage logique des projets (Front, Back)
- Interaction Back-Front

Techno :

- ActiveMq
- Spring boot, Maven
- React, Javascript, AJAX

SUJET

1. PHASE A :

Une entreprise disposant uniquement d'un BackEnd d'application de gestion de cartes (monolithique) souhaite faire évoluer son application.

Cette dernière souhaite mettre en place un dispositif de génération de cartes (image, description, propriétés) à la demande qui sera facturé à l'utilisateur.

Le workflow est le suivant :

- Après s'être connecté, l'utilisateur envoie une demande de création de carte (prompt image, prompt de description)
- Le système reçoit cette demande et fait appel à des services extérieurs (traitement longs >30s) pour la génération des contenus images et textes (e.g. FastSD-CPU, Ollma-qwen2:0.5b)
- Une fois les contenus générés, les propriétés de la carte sont générées (librairie fournie, traitement >30s)
- La carte est ajoutée à l'inventaire de l'utilisateur
- Une somme d'argent (forfaitaire) est débitée à l'utilisateur

Ayant entendu des approches SOA (Service Oriented Application), L'entreprise fait appel à vous afin de transformer son application.

- 1.1. Après avoir analysé et réalisé un workflow de traitement, proposer un diagramme d'architecture répondant aux exigences du SOA (organisation et architecture uniquement) mêlant micro services et ESB.

FAIRE VALIDER VOTRE APPROCHE

- 1.2. Réaliser un tableau récapitulatif des bus de communication les plus répandus (avantages/inconvénients)

1.3. Transformer le BackEnd afin d'accueillir la nouvelle fonctionnalité

FAIRE VALIDER VOTRE APPROCHE

2. PHASE B :

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

2.1. Réaliser un tableau comparatif des principaux Framework FrontEnd existants dressant les avantages et inconvénients de chacun.

Votre choix se porte sur le Framework ReactJS. Votre application devra remplir les fonctionnalités suivantes :

- Une connexion basique de l'utilisateur
- Un écran d'accueil
- La possibilité d'acheter des cartes parmi une liste
- La possibilité de vendre les cartes parmi une liste

2.2. En vous inspirant des spécifications proposées par le client, réaliser un découpage du FrontEnd en composant REACTJS

FAIRE VALIDER VOTRE APPROCHE

2.3. Réaliser votre application REACTJS sans interaction avec le BackEnd dans un premier temps

FAIRE VALIDER VOTRE APPROCH

2.4. Finaliser votre application REACTJS en finalisant les interactions avec le BackEnd

FAIRE VALIDER VOTRE APPROCHE

TIPS

- Penser à versionner vos projets (utiliser gitlab)
- Afin de paralléliser le travail, autoriser le **CROSS ORIGIN** sur vos navigateurs web **ET** sur votre BackEnd
- Utiliser un Bus de communication ActiveMq indépendamment de Springboot.

BONUS

- Utiliser les pipelines de Gitlab pour l'Intégration Continuer (CI), le déploiement continu (CD)
- Utiliser les composants Material-UI de REACT (attention customisation des composants plus complexes)

QUESTIONS

Qu'est ce que le CROSS ORIGIN ? En quoi cela est-il dangereux ?

Comment REACTJS fait-il pour afficher rapidement les modifications sur les composants ?

Quelle est la fonction essentielle de REACTJS ?

Quelle est la fonction essentielle de FLUX ?

Qu'est ce que REDUX ?

Qu'est ce que JMS ? Est-ce spécifique à Springboot ?

Quelles sont les différents modes de transmissions de JMS ?

Quel est le mode de transmission activé par défaut dans activeMq ?

Qu'est ce que activeMq ?

Quels avantages proposent un bus de communication vis-à-vis de requêtes http classiques ?

Comment faire pour partager des modules Maven avec un partenaire extérieur ?

Comment faire pour exporter un composant REACTJS ?

Quel est le pré-requis pour Springboot afin de pouvoir convertir automatiquement le message reçu dans un bus de communication en objet ?

Comment est réalisée la traduction des messages reçus (bus de communication ou request http) en objet Java ? Quelles sont les prérequis ? Est-ce toujours possible ?

Quelles sont les fonctionnalités des différentes annotations en Springboot ?:

- @EnableJpaRepositories
- @EntityScan
- @ComponentScan

REFERENCES

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

[Architecture]

- *** <https://www.oreilly.com/library/view/microservices-vs-service-oriented/9781491975657/>
- ** <https://www.ibm.com/fr-fr/topics/esb>
- ** <https://medium.com/@joudwawad/a-guide-to-communication-styles-in-microservices-architecture-9a8ae4bc21b2>

[Entreprise Service Bus (ESB)]

- *** <https://searchapparchitecture.techtarget.com/definition/Enterprise-Service-Bus-ESB> (5min)
- *** <https://www.oracle.com/technical-resources/articles/middleware/soa-ind-soa-esb.html> (10 min)
- ** https://fr.wikipedia.org/wiki/Enterprise_service_bus
- ** <https://jackynote.medium.com/message-brokers-pros-cons-and-their-crucial-role-in-microservice-3dc6c0df2e53>

[ActiveMq]

- *** <https://activemq.apache.org/getting-started>

[JMS and Springboot]

- *** <https://www.javatodev.com/how-to-use-activemq-with-spring-boot-rest-api>
- ** <https://spring.io/guides/gs/messaging-jms/>

[Communication HTTP]

- *** <https://howtodoinjava.com/spring-webflux/webclient-get-post-example/>
- *** <https://docs.spring.io/spring-framework/reference/web/webflux-webclient/client-builder.html>

[Json to Java Object]

- ** <https://www.mkyong.com/java/jackson-2-convert-java-object-to-from-json/>


[React]


- *** <https://react.dev/learn> (fondamentaux)

ELEMENTS DONNES

- IDE configuré
- 1 sketch
- Une application Monolithique
 - Repo : <https://gitlab.com/js-asi2/asi2-backendmarket-monolithic-student>
 - Server : <http://tp.cpe.fr:8083>
- Une librairie calculant les propriétés d'une carte en fonction d'une image
- 1 Container docker pour la génération de prompt
- 1 Container docker pour la génération d'image
- 1 exemple simple de création d'utilisateur REACT + Redux + Springboot
- Visuel Html UI Semantic

User Form

 Home
Add an user



First Name

First Name

Last Name

Last Name

Password

Your Password


Re-Password


Your Password Again

☐ I agree to the Terms and Conditions

Submit

Login

 Home
Add an user



First Name

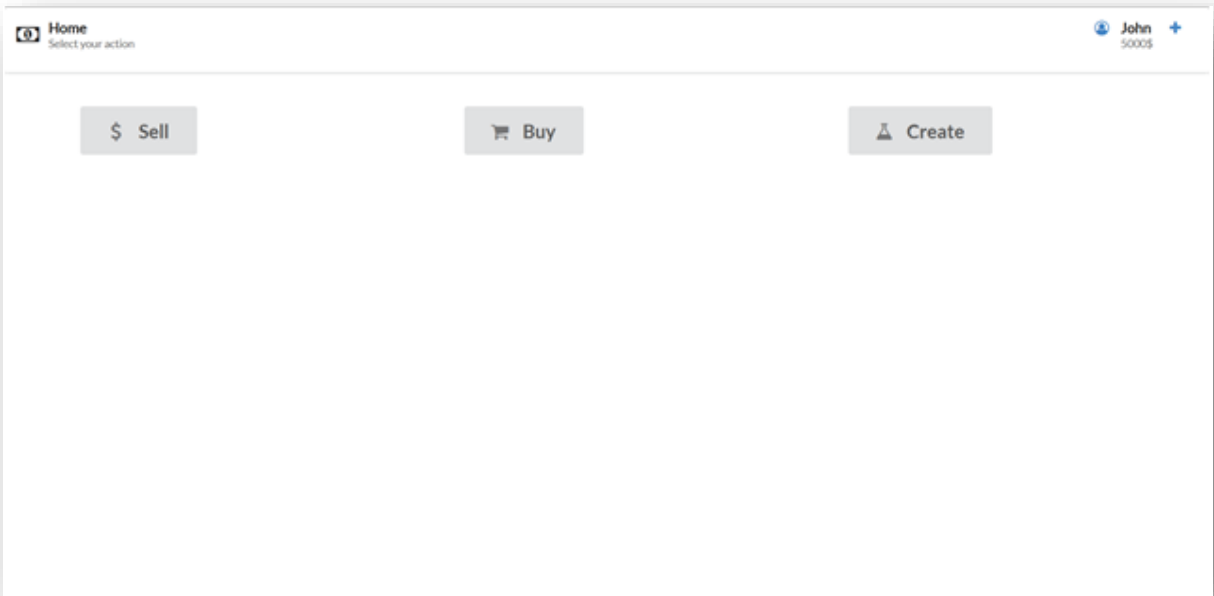
Login

Password

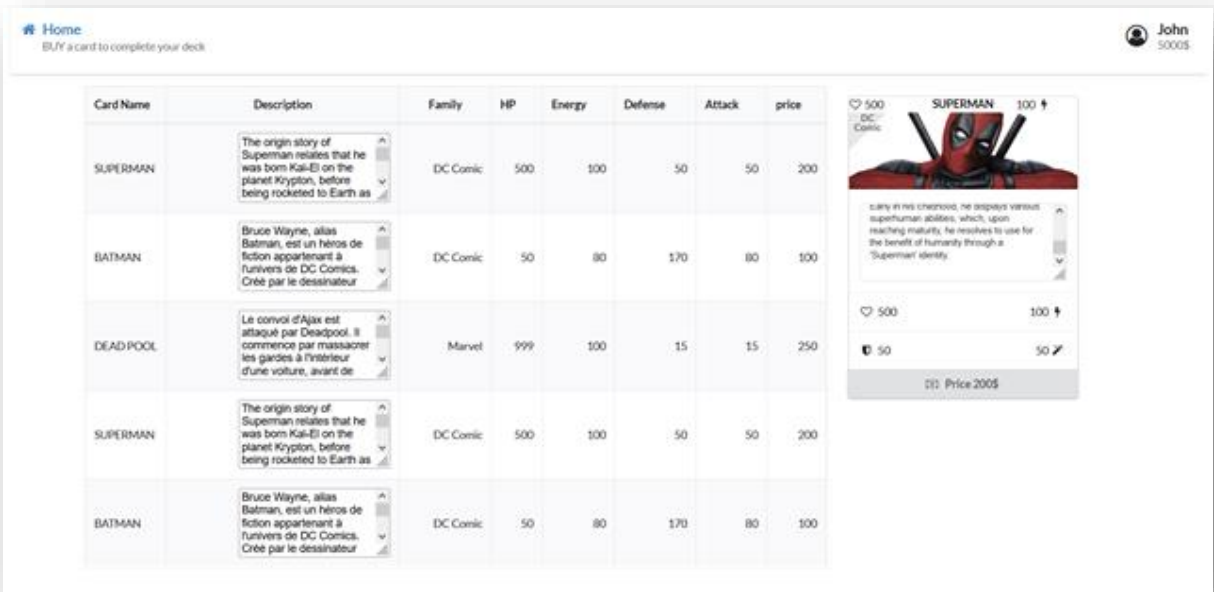
Your Password

Login

Home



Buy



Sell

Home
SELL your cards to get money

John
5000\$

| Card Name | Description | Family | HP | Energy | Defense | Attack | price |
|-----------|---|----------|-----|--------|---------|--------|-------|
| SUPERMAN | The origin story of Superman relates that he was born Kal-El on the planet Krypton, before being rocketed to Earth as | DC Comic | 500 | 100 | 50 | 50 | 200 |
| BATMAN | Bruce Wayne, alias Batman, est un héros de fiction appartenant à l'univers de DC Comics. Créé par le dessinateur | DC Comic | 50 | 80 | 170 | 80 | 100 |
| DEAD POOL | Le convoi d'Ajax est attaqué par Deadpool. Il commence par massacrer les gardes à l'intérieur d'une voiture, avant de | Marvel | 999 | 100 | 15 | 15 | 250 |
| SUPERMAN | The origin story of Superman relates that he was born Kal-El on the planet Krypton, before being rocketed to Earth as | DC Comic | 500 | 100 | 50 | 50 | 200 |
| BATMAN | Bruce Wayne, alias Batman, est un héros de fiction appartenant à l'univers de DC Comics. Créé par le dessinateur | DC Comic | 50 | 80 | 170 | 80 | 100 |

500
DC
Comic

SUPERMAN

100

The origin story of Superman relates that he was born Kal-El on the planet Krypton, before being rocketed to Earth as an infant by his scientist father Jor-El, moments before Krypton's destruction. Discovered and adopted by a Kryptonian scout, he

500100

5050

Price 200\$

Create

Home
Generate a card

John
5000\$

Image Prompt

Description Prompt

☐ I agree to the Terms and Conditions

Generate

999
Marvel

DEAD POOL

100

Le convoi d'Ajax est attaqué par Deadpool. Il commence par massacrer les gardes à l'intérieur d'une voiture, avant de s'en prendre au reste du convoi. Après une longue escarmouche, où il est

999100

1515