

ALOM - Seconde Session

Table of Contents

1. Contraintes de la session	1
1.1. Repository GitLab de rendu	1
1.2. Contraintes techniques	2
2. Services mis à disposition	2
3. Sujet	3
3.1. En tant que dresseur, je peux ouvrir un booster de cartes, contenant 5 cartes aléatoires.	3
3.2. En tant que dresseur, je ne peux ouvrir que 2 boosters par jour	4
3.3. En tant que dresseur, après avoir ouvert un booster de cartes, je choisis un Pokemon pour l'ajouter à mon équipe	5

Session de rattrapage master MIAGE - Groupe 1

Pokemon TCG Pocket est un jeu de cartes à collectionner. Les dresseurs de Pokemon piochent tous les jours 2 boosters de 5 cartes, pour constituer leur collection. Ces cartes représentent chacune un Pokemon.

L'objectif de ce sujet est de créer une application *booster*, qui permet à un dresseur de tirer 2 boosters par jour, de sélectionner une carte dans chaque booster, et d'ajouter les Pokemon à son équipe.

1. Contraintes de la session

La session se déroule ce vendredi 7 mars 2025, de 13h30 à 17h30.

1.1. Repository GitLab de rendu

Créez votre repository GitLab avec ce lien : <https://gitlab-classrooms.cleverapps.io/assignments/820d0a29-5478-40c3-a59f-687f85d61c51/accept>

Il contient un squelette d'application que vous pouvez utiliser.

Le rendu se fait via vos repository GitLab.

Vous devez push votre code directement sur le repository qui vous est affecté, et que vous avez créé dans GitLab Classrooms.



Évitez de commiter/pusher vos répertoires **target** et vos fichiers eclipse/intelliJ **.idea**, **.settings**... Vous devez déjà avoir un fichier **.gitignore** à la racine de vos projets pour cela.



Les repositories GitLab seront fermés automatiquement le vendredi 7 mars 2025 à 18h.

1.2. Contraintes techniques

Les fonctionnalités demandées sont exposées à travers une API REST/JSON. Il n'est pas demandé le développement d'IHM pour ces fonctionnalités.

Les données de votre API doivent être stockées dans une base de données **h2**, en utilisant JPA.

Pour la configuration d'une base de données h2, vous pouvez vous reporter au [TP sur la persistance des données](#).

L'écriture de tests unitaires avec JUnit n'est pas obligatoire.

Il n'est pas nécessaire de déployer votre code sur Clever Cloud, seul le rendu GitLab est important.

Il n'est pas nécessaire non plus de sécuriser votre API avec Spring Security.

2. Services mis à disposition

Pour vous aider à développer, des services *Pokemon Type API* et *Trainer API* ont été déployés sur Clever Cloud et sont accessibles. Vous devrez vous connecter à ces services pour lister les Pokemons, et mettre à jour les Trainer.

Les services sont accessibles aux URLs suivantes :

- Pokemon Type API : <https://pokemon-types.cleverapps.io/>
 - GET <https://pokemon-types.cleverapps.io/pokemon-types/>
 - GET <https://pokemon-types.cleverapps.io/pokemon-types/25>
- Trainer API : <https://trainers.cleverapps.io/>
 - GET <https://trainers.cleverapps.io/trainers/>
 - GET <https://trainers.cleverapps.io/trainers/Ash>
 - GET <https://trainers.cleverapps.io/trainers/Misty>

Pour Trainer API, un Swagger est aussi disponible : <https://trainers.cleverapps.io/swagger-ui/index.html>

Vous pouvez utiliser ce Swagger pour créer un Trainer vous appartenant avec une requête POST, voici un exemple :

```
curl -X 'POST' \  
  'https://trainers.cleverapps.io/trainers/' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "name": "Major Bob",
```

```

"email": "major-bob@pokemon-champions.org",
"team": [
  {
    "pokemonTypeId": 100,
    "level": 21
  },
  {
    "pokemonTypeId": 25,
    "level": 18
  },
  {
    "pokemonTypeId": 26,
    "level": 24
  }
]
}'

```

L'ajout d'un Pokemon à un Trainer se fait avec une requête POST :

```

curl -X 'POST' \
  'https://trainers.cleverapps.io/trainers/Ash/catchPokemon' \
  -H 'Content-Type: application/json' \
  -d '{
    "pokemonTypeId": 141,
    "level": 5
  }'

```

3. Sujet

Le sujet est composé de *User Stories*. L'implémentation que vous faites des *User Stories* est libre, charge à vous d'implémenter une architecture de code qui vous semble correcte, en vous inspirant de ce que nous avons déjà implémenté en TP.

3.1. En tant que dresseur, je peux ouvrir un booster de cartes, contenant 5 cartes aléatoires.

L'ouverture d'un booster doit se faire avec une requête de ce type, avec en paramètre l'identifiant du Trainer qui ouvre le booster :

```

GET /booster?trainerId=Ash

{
  "boosterId": "a5338850-ece2-4c0c-aa67-77b825f3ee70",
  "trainerId": "Ash",
  "openingDate": "2025-03-07",
  "cardSelected": false,

```

```

"cards": [
  { "cardId": "2e66584b-c794-4a08-84be-8f4cb98baf66", "pokemonTypeId": 25, "name":
"Pikachu", "level": 6 },
  { "cardId": "a412285b-e76b-4667-b9fe-55e0a5068262", "pokemonTypeId": 12, "name":
Butterfree", level": 1 },
  { "cardId": "d8020868-0fdd-4e77-b9a5-2079e2f987f1", "pokemonTypeId": 65, "name":
"Alakazam", level": 60 },
  { "cardId": "3bfe98b8-7b79-4a6f-a204-1521ac0d4094", "pokemonTypeId": 25, "name":
"Pikachu", "level": 13 },
  { "cardId": "3ab50e7b-2fdc-4842-ae4e-acdef1f745d6", "pokemonTypeId": 125, "name":
"Electabuzz", "level": 44 }
]
}

```

Les boosters sont stockés en base de données après avoir été générés aléatoirement lors de l'appel.

Chaque booster a un identifiant unique (qui peut être un numéro de séquence de type **Long**, ou un **UUID**).

La date "openingDate" correspond à la date de création du booster.

La valeur "cardSelected" permet d'indiquer que le booster a été ouvert, mais qu'aucune carte n'a été choisie dans le booster pour l'instant.

Les cartes sont aléatoires, avec les identifiants de Pokemon ayant une valeur entre 1 et 151.

Une carte avec le même Pokémon peut apparaître plusieurs fois (comme Pikachu dans l'exemple).

Chaque carte a un identifiant unique (qui peut être un numéro de séquence de type **Long**, ou un **UUID**).

Le niveau de la carte est un nombre aléatoire entre 1 et 50.

La génération d'un nombre aléatoire peut se faire avec **Random.nextInt()**



Pour la récupération d'un "name" de la carte, un appel à *Pokemon Type* peut s'avérer utile.

3.2. En tant que dresseur, je ne peux ouvrir que 2 boosters par jour

L'ouverture de 2 boosters sur la même journée est possible.

L'ouverture d'un troisième booster renvoie une erreur avec un statut HTTP **402 Payment Required**.

Le champ "openingDate" des boosters permet de vérifier le nombre de boosters ouvert par le dresseur sur une journée donnée.

3.3. En tant que dresseur, après avoir ouvert un booster de cartes, je choisis un Pokemon pour l'ajouter à mon équipe

La sélection d'une carte dans un booster se faire avec la requête suivante :

```
POST /booster/{boosterId}/chooseCard/{cardId}
```

En reprenant l'exemple précédent, une requête correcte serait

```
POST /booster/a5338850-ece2-4c0c-aa67-77b825f3ee70/chooseCard/a412285b-e76b-4667-b9fe-55e0a5068262
```

Si une carte a déjà été sélectionnée dans le booster (`cardSelected == true`), retourner une erreur 400.

Si la carte choisie n'existe pas dans le booster, retourner une erreur 404.

Si toutes les conditions sont remplies, le Pokemon peut être ajouté à l'équipe du dresseur qui a ouvert le booster, avec une requête sur l'API Trainer, en passant en paramètre l'identifiant du Pokemon et le level de la carte sélectionnée :

```
curl -X 'POST' \
  'https://trainers.cleverapps.io/trainers/Ash/catchPokemon' \
  -H 'Content-Type: application/json' \
  -d '{
    "pokemonTypeId": 125,
    "level": 44
  }'
```

Si tout est ok, le booster est validé et la `cardSelected` doit être positionné à `true` en base de données.

Fin du sujet