

(Java/Spring boot)

API Dessin

Enoncé

Ce projet est une petite application permettant de gérer des dessins en ligne.

Les informations du dessin sont:

Information	Type	Description
id	numeric	Identifiant unique du dessin
name	string	nom du dessin
width	numeric	Largeur du dessin
height	numeric	Hauteur du dessin
description	string	Description du dessin

Exemple de la représentation d'un dessin en JSON

```
{
  "id" : 1235,
  "name" : "dessin001",
  "width" : 1000,
  "height" : 900,
  "description": "définition d'un exemple de dessin"
}
```

Exemple de la representation d'un tablea de dessins en JSON

```
[
  {
    "id": 1235,
    "name": "dessin001",
    "width": 1000,
    "height": 900,
    "description": "definition d'un exemple de dessin"
  },
  {
    "id": 456,
    "name": "dessin002",
    "width": 1000,
    "height": 900,
    "description": "definition d'un autre exemple de dessin"
  }
]
```

Il n'y a pas la partie graphique (pas d'IHM) mais une API (REST) qui permettra de:

- Créer un dessin. **méthode HTTP POST**
 - Le contenu du dessin sera dans le BODY du message.
- Sauvegarder un dessin. **Méthode HTTP PUT.**
 - Le contenu du dessin sera dans le BODY du message.
- Lister les dessins. **Méthode HTTP Get.**
 - Aucun paramètre ou information attendu en entrée du serveur.
- Récupérer un dessin via son ID. **Méthode HTTP GET.**
 - L'ID du dessin sera récupéré via queryString ex:
http://localhost:8080/dessin?id=1234
- Récupérer un dessin via son Nom. **Méthode HTTP GET.**
 - Le nom du dessin sera récupéré via le path ex:
http://localhost:8080/dessin/dessin001

Le modèle entrant/sortant des verbes de l'api est **application/json**.

La persistance des données sera réalisée dans une base de données (h2) en utilisant JPA.

Attendus

1. Conception

Création du projet avec spring "Spring initializr".

Les caractéristiques sont:

- JDK 11
- Maven
- Version stable de spring boot (2.7.X).

2. Implémentation

Il est attendu les éléments suivants:

- Au moins un test unitaire.
- Au moins un contrôleur pour la gestion de la communication HTTP.
- Un repository pour la persistance du type Dessin.
 - Avec les finders findById, findByName, findAll.
- **Utilisation systématique du logger** et pas de System.out ou System.err .
- Un code structuré et organisé (gestion des packages)
- Utilisation de Lombok (tools) si nécessaire.
- Un **pom.xml** avec les dépendances nécessaires.

Remise du projet:

Chaque personne à un repository créé sur github. La remise de cet exercice sera réalisée via ce projet github.

Pour info la liste des repository est la suivante (le discriminant est le prénom)

- https://github.com/aduglas/spv_gilles
- https://github.com/aduglas/spv_benjamin
- https://github.com/aduglas/spv_janeka
- https://github.com/aduglas/spv_thierno
- https://github.com/aduglas/spv_rowan
- https://github.com/aduglas/spv_ilker
- https://github.com/aduglas/spv_antoine
- https://github.com/aduglas/spv_laurent
- https://github.com/aduglas/spv_clement
- https://github.com/aduglas/spv_jerome
- https://github.com/aduglas/spv_alexandre
- https://github.com/aduglas/spv_maximilien

La remise du projet est attendue au maximum le 22/11/2022.

Je vous demande de bien vouloir signaler la remise de votre projet à une des adresses mail suivantes:

- duglasalain@gmail.com
- formateur_alain.duglas@supdevinci-edu.fr