Cave à vins (Partie 07)

# TP01 du module 04 – Spring Web Service

|  |
| --- |
| Ce TP sera un TP fil rouge du cours   * Configurer le projet avec le starter Spring Web * Créer un contrôleur REST * Créer une méthode GET |

|  |
| --- |
| **Durée estimée** |
| 30 minutes |

Contexte

En vous appuyant sur l’ensemble des concepts abordés dans le cours, mettez en place une application Spring Boot permettant la gestion d’une cave à vins.

L’application de cave à vin sera une application d’API Web sécurisée à la fin des itérations :

* Elle permet de gérer le stock des bouteilles de vins par le propriétaire qui est un administrateur de l’application
* Elle permet de gérer l’achat en ligne des bouteilles par un client
  + Les bouteilles de vin peuvent avoir 3 couleurs : Rouge, Blanc ou Rosé
  + Les bouteilles de vin sont associées aux régions françaises
* Un client et un propriétaire doivent s’authentifier
* Un visiteur est un client qui n’a pas de compte
* Il pourra uniquement consulter le stock

Dans les itérations nous réaliserons uniquement le back de l’application.

Cette application se fera avec Spring Boot.

Dans les itérations de ce module, nous allons développer l’API Web.

* Si vous n’aviez pas fini le TP du module 02. Vous trouverez les classes solutions dans un zip des ressources.

## Voici le diagramme des cas d’utilisation complet :

Une image contenant texte, diagramme

Description générée automatiquement

## Voici le diagramme complet des entités :

Une image contenant texte, diagramme, Parallèle, Plan

Description générée automatiquement

Enoncé

# Copier la couche BLL fournie

Pour faire gagner du temps, et vous concentrer sur les contrôleurs ; nous vous fournissons une partie de la couche BLL

* Vous trouverez l’interface BouteilleService et son implémentation dans les ressources
* Copier les dans fr.eni.cave.bll
* Voici le diagramme des classes :

Une image contenant texte, capture d’écran, Police, noir et blanc

Description générée automatiquement

* Les méthodes charger sont pour les profils visiteur, Client et Proprio
* Les méthodes ajouter (= ajouter ou modifier) ou supprimer sont pour le Proprio

## Modifier BouteilleRespository

* Pour faire fonctionner le code, il faut lui ajouter la méthode :

// Rechercher une bouteille par son nom

Bouteille findByNom(@Param("nom") String nom);

# Configurer le projet en projet Web

# Mettre le mode update pour la base

# Créer le contrôleur BouteilleController

* L’URL racine est : /caveavin/bouteilles
* Injecter lui le service
* Définissez une méthode Get pour retourner l’ensemble des bouteilles
  + Si la liste est vide retourner un statut 204

# Tester avec Postman

* Lancer l’application
* Votre base de données existe et les tables aussi

## Test sans donnée en base

* + Faire un premier test, il n’y a pas de données en base donc un statut 204

## Test avec des données en base

* Utiliser le fichier de script SQL (script\_bouteilles.sql) fourni pour mettre des données en base
* Valider le comportement de votre contrôleur