Cave à vin (partie 01)

# TP01 du module 02 – Spring Data

|  |
| --- |
| Ce TP sera un TP fil rouge du cours   * Mise en place de Spring Boot et Spring Core * Création des premières entités JPA * Création des premiers Repository de Spring Data JPA * Utilisation de tests unitaires pour valider |

|  |
| --- |
| **Durée estimée** |
| 30 minutes et 1 heure |

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 vin.

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.

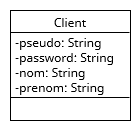
Cette application se fera avec Spring Boot.

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

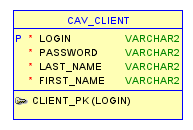
Une image contenant texte, diagramme

Description générée automatiquement

## Voici le diagramme de l’entité à réaliser :



Voici le diagramme de la table en base de données :

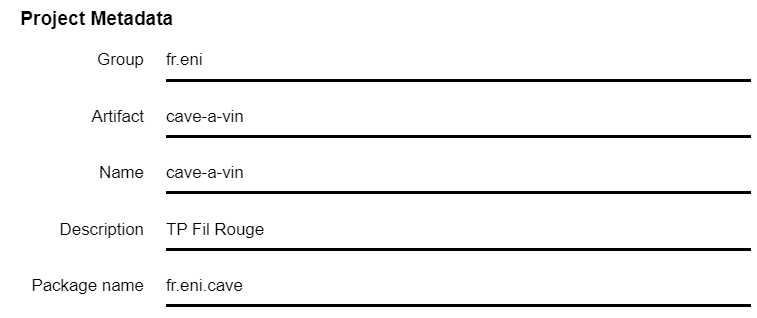


Énoncé

# Créer une application CaveAVin avec SpringBoot

* Utiliser Spring initializr et sélectionner :
* Gradle (privilégier avec le langage Groovy)
* Java en version minimum 17
* La version courante de Spring Boot (Release)
* Le packaging est en Jar

## Métadonnées :



## Dépendances Spring Boot :

* Lombok
* Spring Boot DevTools
* Spring Data JPA
* MS SQL Server Driver
* Ajouter la dépendance vers la base de tests H2 dans build.gradle :

//Ajout de la dépendance de la datasource H2 pour les tests unitaires sur JPA

testImplementation 'com.h2database:h2'

# Configurer la base de données

* Créer une base de données appelée : CAVE\_A\_VIN
* Configurer la source de données associées

# Créer l’entité Client

* Le package de la classe est fr.eni.cave.bo.client
* Utiliser les annotations de JPA
* Utiliser les annotations de Lombok
  + Utiliser uniquement l’attribut pseudo pour la méthode equals
  + Empêcher que le mot de passe soit tracer

# Créer ClientRepository

# Créer des tests unitaires

* Créer une classe de tests unitaires annotées @DataJpaTest
* Créer 2 tests :
  + Sauver un client
  + Supprimer un client
  + Vos tests doivent tracer le client pour valider que le mot de passe ne s’affiche pas
  + Et utiliser au moins une fois le equals pour valider le comportement
* Voici le type de trace désiré :

Client(pseudo=bobeponge@email.fr, nom=Eponge, prenom=Bob)