

_			•		. •			
ı 1	Δ		rı	n	tı	$\mathbf{a}$	n	•
$\boldsymbol{\nu}$	es	,,,		μ	LI	v		٠

Devoir POO Génie Logiciel — partie backend

## Technos:

•Stack: Spring Boot (REST API)

•Base de données : MariaDB ou MySQL

## Prérequis:

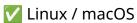
- Git
- •Java 17+
- •Maven (facultatif, car Maven Wrapper est inclus)
- •Une base de données MySQL ou MariaDB

## **Initialisation:**

1.Cloner le projet :

git clone https://github.com/haritsimba/PooGLog.git cd PooGLog

2.Créer un fichier application.yaml dans le dossier src/main/resources/ et Copier les contenus de src/main/resources/application.yaml.example dans src/main/resources/application.yaml



H Windows
copy src\main\resources\application.yaml.example src\main\resources\application.yaml  Linux / macOS
chmod +x mvnw ./mvnw clean install
3.Ajuster src/main/resources/application.yaml.example selon vos configurations 4.Installer les dépendances :
₩ Windows
mvnw.cmd clean install 5.Lancer votre projet :
✓ Linux / macOS
./mvnw spring-boot:run  Windows
mvnw.cmd spring-boot:run
Structure du projet :
PooGLog/
— poo/       — configurations/ # Configuration des beans et autres       — controllers/ # Contrôleurs Spring (API REST)       — dtos/ # Objets de transfert de données       — entities/ # Entités JPA pour la base de données       — PooApplication.java # Point d'entrée de l'application Spring Boot       — repositories/ # Interfaces pour l'accès aux données (JPA)       — services/ # Logique métier de l'application     — utils/ # Fonctions utilitaires     — enumerations/ # Énumérations     — resources/     — application.yaml # Config personnalisée     — application.yaml.example # Exemple de config   — test/ # Tests unitaires
— poo/       — configurations/ # Configuration des beans et autres       — controllers/ # Contrôleurs Spring (API REST)       — dtos/ # Objets de transfert de données       — entities/ # Entités JPA pour la base de données       — PooApplication.java # Point d'entrée de l'application Spring Boot       — repositories/ # Interfaces pour l'accès aux données (JPA)     — services/ # Logique métier de l'application     — utils/ # Fonctions utilitaires     — enumerations/ # Énumérations     — resources/     — application.yaml # Config personnalisée     — application.yaml.example # Exemple de config   — test/ # Tests unitaires   — mvnw, mvnw.cmd # Maven Wrapper (Windows & Linux)   — pom.xml # Dépendances du projet

## **Références:**

- •Pour ceux qui ne sont pas familiarisés avec Spring Boot, je vous recommande cette chaîne YouTube
- •Pour ceux qui n'ont jamais utilisé lombok : <u>pour éviter les tâches répétitives</u> (<u>constructeur,getters,setters,...</u>)
- •Rélation dans spring data jpa