

# **Microservices-Developpeurs**

## **Travaux Pratiques**



zenika  
ARCHITECTURE INFORMATIQUE

# Pré-requis

## Installation

---

Installer les technos demandées

- docker
- docker-compose
- java 8 (JDK)
- maven 3.2 ou supérieur

# Use Case

## Objectifs

---

Réaliser un projet Java pour assurer la persistance du modèle métier de l'application RésaNet.

Ce premier exercice a pour objectif la mise en place du projet dans votre IDE, et l'écriture d'un premier fichier de mapping pour manipuler des instances de la classe `Ville`.

## Modèle

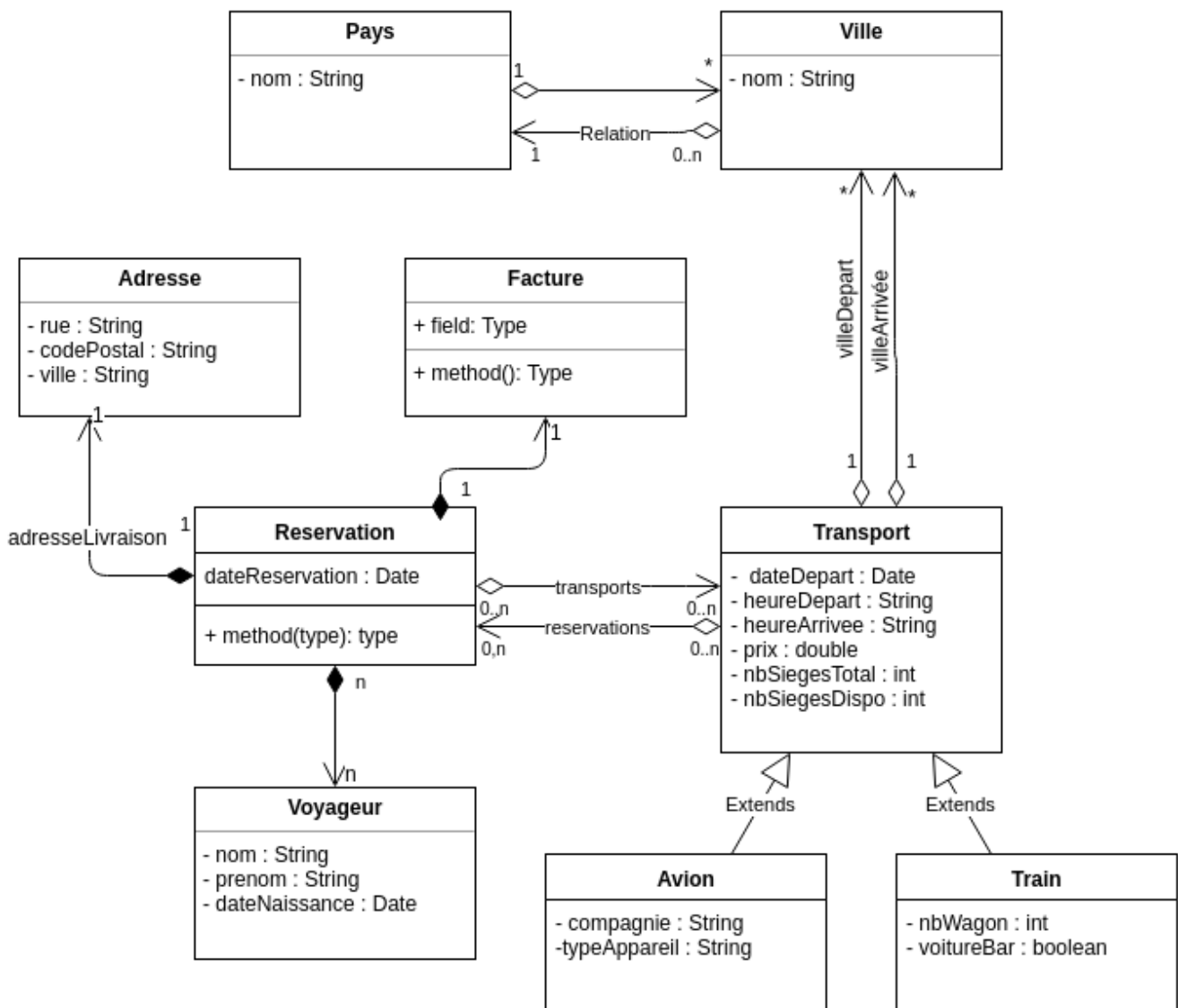
---

Une réservation consiste en une ville de départ, une ville d'arrivée, un ou plusieurs moyens de transports et des voyageurs.

Pour être complet, une facture ainsi qu'une adresse de livraison seront associées à une réservation.

Les transports peuvent être de 2 types :

- Avion
- Train



## Mise en place du projet

- Démarrer Eclipse
- Depuis Eclipse, importer le projet fourni par le formateur
  - File > Import > Maven > Existing Maven Projects

# TP-00 Prise en Main de l'environnement

Estimated Time **20 minutes**

## Explications

---

L'environnement est composé de :

- Spring-Boot
- Spring-Data-Jpa
- Hibernate comme implémentation de JPA
- Une base de données H2 en mode TCP

## Faire un check des dépendances

---

Dans `pom.xml` vérifier les dépendances et leur correspondance à l'environnement décrit.

## Configuration Spring-Boot

---

Regarder dans `com.zenika.microservices.resanet.starter.Application.java`

Cette classe contient tout ce qu'il faut pour lancer l'application

```
@EnableSwagger2
@SpringBootApplication(scanBasePackages={"com.zenika.microservices.resanet.controller",
    "com.zenika.microservices.resanet.services"})
@EnableJpaRepositories(basePackages={"com.zenika.microservices.resanet.repository"})
@EntityScan(basePackages={"com.zenika.microservices.resanet.domain"})
public class Application {

    public static void main(final String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Lancement de l'application :

```
mvn spring-boot:run
```

## Check de la base de données

---

Au lancement de l'application une base de données H2 est disponible.

On peut la browser sur son port par défaut 8082

voir dans applications.properties pour l'url de la db :

```
jdbc:h2:tcp://localhost/~/.resanet
```

Vérifier que le fichier import.sql popule bien la bdd

## Voir la configuration Hibernate

---

Dans `com.zenika.microservices.resanet.domain` -> regarder les classes annotées qui correspondent à une application monolithique classique.

## Check de la configuration REST

---

Depuis le browser demander la listes des villes :

`http://localhost:8000/villes`

**\*\* Fin du TP-00 \*\***

