ECOLE NORMALE SUPÉRIURE DE L'ENSEIGNEMENT
TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA

Examen Architecture JEE et Middlewares

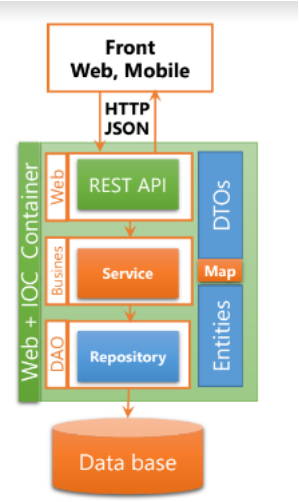
Réalisé par :
AKASMIOU Ouassima
Glsid 2

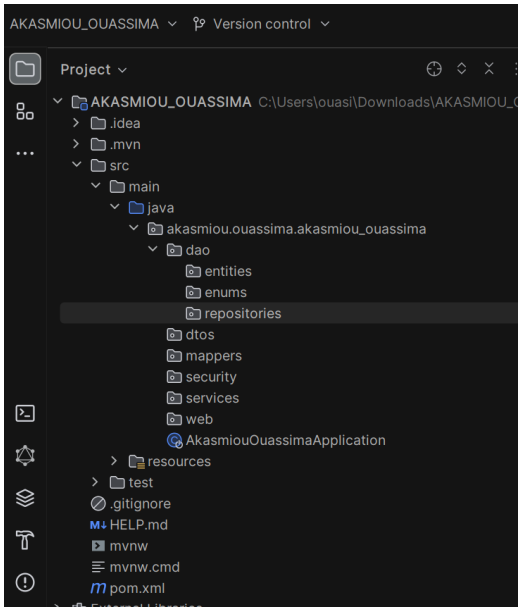
Professeur :
Mr Mohamed Youssfi

Année universitaire 2022-2023

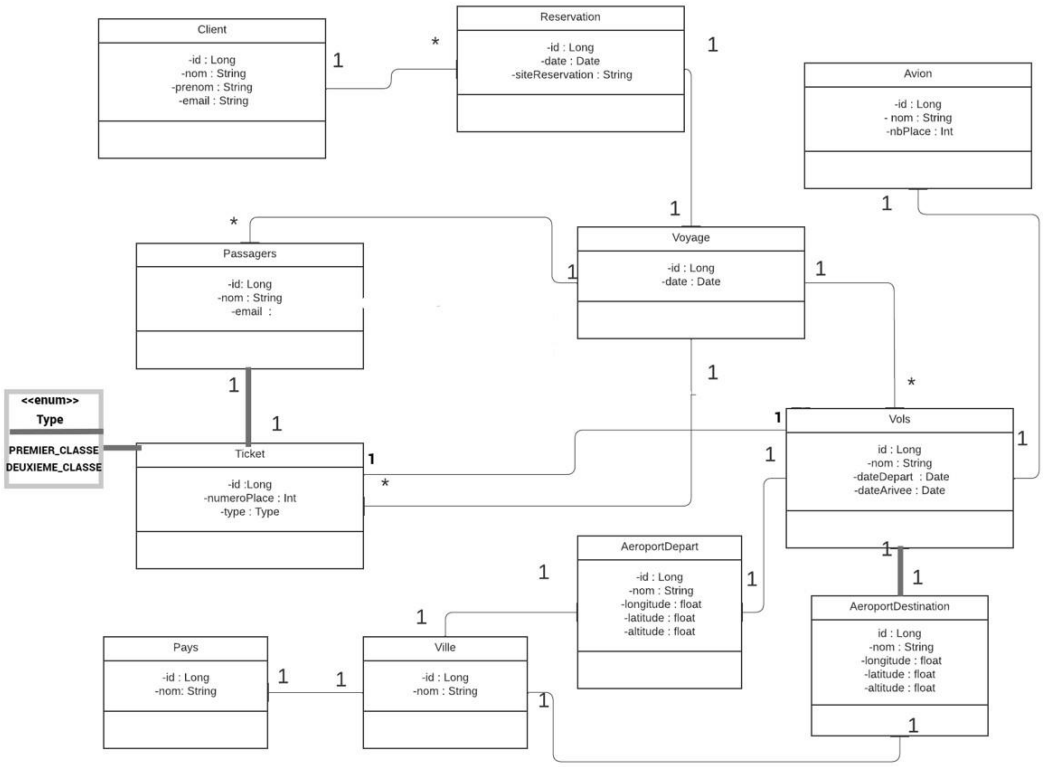
A. Conception :

1. Établir une architecture technique du projet



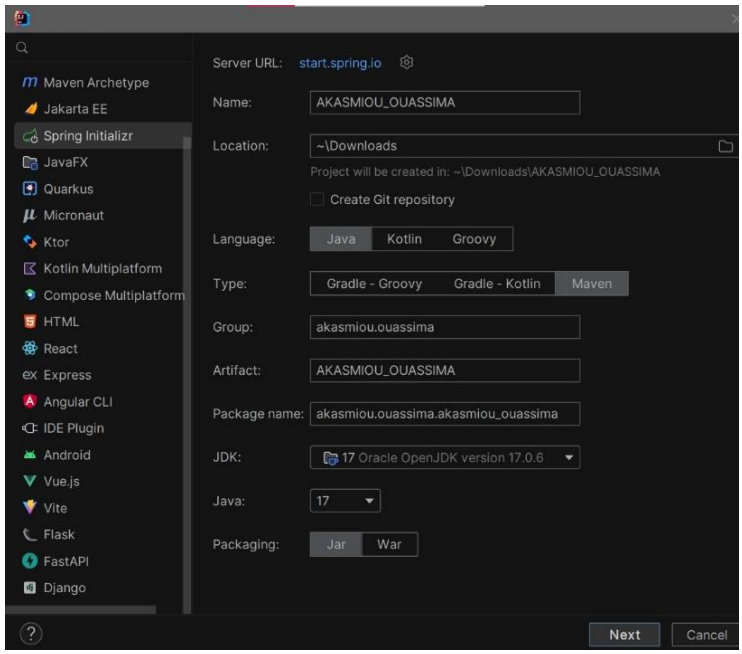


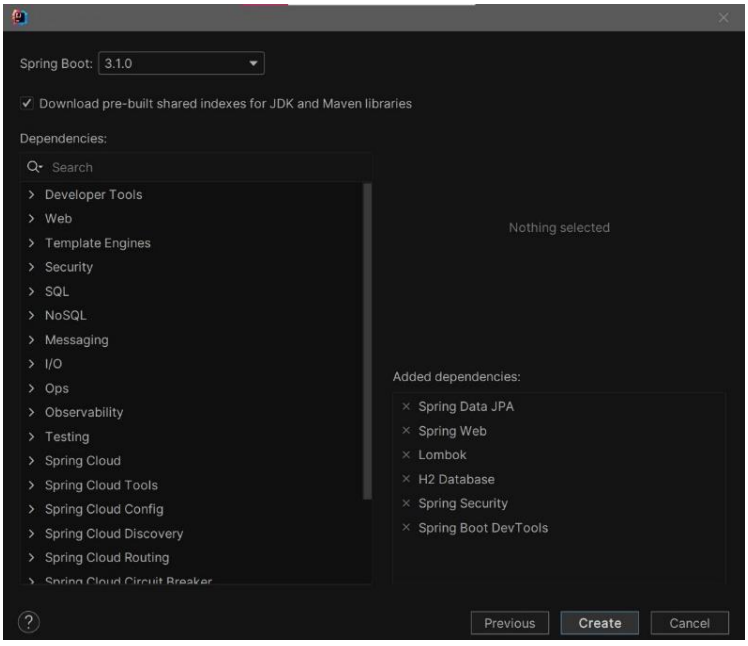
2. Établir un diagramme de classes qui montre les entités. On ne représentera que les attributs.



B. Implémentation :

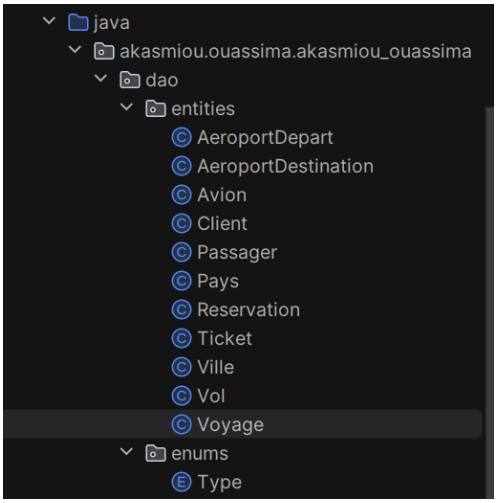
1. Créer un projet Spring boot avec les dépendances requises. Les identifiants du projet GroupId, ArtifactId et le package de base doivent contenir votre nom et prénom.





2. Couche DAO

a. Créer les entités JPA



Voyage :

```java
package akasmiou.ouassima.akasmiou ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.format.annotation.DateTimeFormat;

import java.util.Date;
import java.util.List;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Voyage {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date date;
    @OneToMany(mappedBy = "voyage")
    private List<Passager> passagers;
    @OneToOne
    private Reservation reservation;
    @OneToMany(mappedBy = "voyage")
    private List<Ticket> tickets;
    @OneToMany(mappedBy = "voyage")
    private List<Vol> vols;
}
```

Vol :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.format.annotation.DateTimeFormat;

import java.util.Date;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Vol {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date dateDepart;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date dateArrivee;

    @OneToOne(mappedBy = "vol" )
    private  AeroportDepart aeroportDepart;
    @OneToOne(mappedBy = "vol" )
    private  AeroportDestination aeroportDestination;
    @OneToOne
    private  Avion avion;
    @OneToOne
    private Ticket ticket;
    @ManyToOne(fetch = FetchType.LAZY)
    private Voyage voyage;
}
```

Ville :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Ville {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    @OneToOne(mappedBy = "ville" )
```

```
    private  AeroportDepart aeroportDepart;
    @OneToOne(mappedBy = "ville" )
    private  AeroportDestination aeroportDestination;
    @OneToOne(mappedBy = "ville" )
    private  Pays pays;
}
```

Ticket :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import akasmiou.ouassima.akasmiou_ouassima.dao.enums.Type;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Ticket {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private int numeroPlace;
    @Enumerated(EnumType.STRING)
    private Type type;
    @OneToOne
    private Passager passager;
    @OneToOne(mappedBy = "ticket")
    private Vol vol;
    @ManyToOne(fetch = FetchType.LAZY)
    private Voyage voyage;

}
```

Enum Type :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.enums;

public enum Type {
    PREMIERE_CLASSE,DEUXIEME_CLASSE
}
```

Reservation :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.format.annotation.DateTimeFormat;

import java.util.Date;

@Entity
@Data
@AllArgsConstructor
```

```java
@NoArgsConstructor
public class Reservation {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date date;
    private String siteReservation;
    @ManyToOne(fetch = FetchType.LAZY)
    private Client client;
    @OneToOne(mappedBy = "reservation")
    private Voyage voyage;
}
```

Pays :

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Pays {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    @OneToOne()
    private Ville ville;
}
```

Passager :

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Passager {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private String email;
    @ManyToOne
    private Voyage voyage;
    @OneToOne(mappedBy = "passager")
```

```
    private Ticket ticket;
}
```

Client :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.List;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Client {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private String prenom;
    private String email;
    @OneToMany(mappedBy = "client")
    private List<Reservation> reservations;

}
```

Avion :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Avion {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private int nbPlace;
    @OneToOne(mappedBy = "avion")
    private Vol vol;
}
```

AeroportDestination :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class AeroportDestination {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private float longitude;
    private float latitude;
    private float altitude;
    @OneToOne
    private Vol vol;
    @OneToOne
    private Ville ville;

}
```

AeroportDepart :

```
package akasmiou.ouassima.akasmiou_ouassima.dao.entities;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class AeroportDepart {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private float longitude;
    private float latitude;
    private float altitude;
    @OneToOne
    private Vol vol;
    @OneToOne
    private Ville ville;
}
```

b. Créer les interfaces JPA Repository basées sur Spring Data

```
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.AeroportDepart;
import org.springframework.data.jpa.repository.JpaRepository;

public interface AeroportDepartRepository extends
JpaRepository<AeroportDepart, Long> {
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.AeroportDestination;
import org.springframework.data.jpa.repository.JpaRepository;

public interface AeroportDestinationRepository extends
JpaRepository<AeroportDestination, Long> {
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Avion;
import org.springframework.data.jpa.repository.JpaRepository;

public interface AvionRepository extends JpaRepository<Avion, Long> {
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Client;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ClientRepository extends JpaRepository<Client, Long> {
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Passager;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PassagerRepository extends JpaRepository<Passager, Long> {
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Pays;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PaysRepository extends JpaRepository<Pays, Long> {
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Reservation;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ReservationRepository extends JpaRepository<Reservation,
Long> {
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Ticket;
import org.springframework.data.jpa.repository.JpaRepository;

public interface TicketRepository extends JpaRepository<Ticket, Long> {
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Ville;
import org.springframework.data.jpa.repository.JpaRepository;

public interface VilleRepository extends JpaRepository<Ville, Long> {
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Vol;
import org.springframework.data.jpa.repository.JpaRepository;

public interface VolRepository extends JpaRepository<Vol, Long> {
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dao.repositories;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Voyage;
import org.springframework.data.jpa.repository.JpaRepository;

public interface VoyageRepository extends JpaRepository<Voyage, Long> {
}
```

c. Tester la couche DAO avec une application qui alimente la base de données avec quelques enregistrements de test.

```
package akasmiou.ouassima.akasmiou_ouassima;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.*;
import akasmiou.ouassima.akasmiou_ouassima.dao.repositories.*;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.stream.Stream;

@SpringBootApplication
public class AkasmiouOuassimaApplication {

    public static void main(String[] args) {
        SpringApplication.run(AkasmiouOuassimaApplication.class, args);
    }
    @Bean
    CommandLineRunner start(AeroportDepartRepository
aeroportDepartRepository,
                            AeroportDestinationRepository
aeroportDestinationRepository,
                            ClientRepository clientRepository,
                            ReservationRepository reservationRepository,
                            VolRepository volRepository,
                            VoyageRepository voyageRepository,
                             VilleRepository villeRepository,
                            PaysRepository paysRepository,
                             PassagerRepository passagerRepository,
                            TicketRepository ticketRepository,
                            AvionRepository avionRepository
                            ){
        return args -> {
                List<Client> clients = new ArrayList<>();
                Stream.of("ouassima","mohamed","jinan").forEach(name ->{
                    Client client = new Client();
                    client.setNom(Math.random()<0.5? "akasmiou": "khladi");
                    client.setPrenom(name);
                     client.setEmail(name+"@gmail.com");
                     clientRepository.save(client);
                    clients.add(client);
                        });


            Reservation reservation1 = new Reservation();
            reservation1.setDate(new Date());
            reservation1.setSiteReservation("www.reservation.com");
            reservation1.setClient(clients.get(0));
            reservationRepository.save(reservation1);
            Reservation reservation2 = new Reservation();
            reservation2.setDate(new Date());
            reservation2.setSiteReservation("www.reservation.com");
```

```
        reservation2.setClient(clients.get(1));
        reservationRepository.save(reservation2);
        Reservation reservation3 = new Reservation();
        reservation3.setDate(new Date());
        reservation3.setSiteReservation("www.reservation.com");
        reservation3.setClient(clients.get(2));
        reservationRepository.save(reservation3);

        Voyage voyage1 = new Voyage();
        voyage1.setDate(new Date());
        voyage1.setReservation(reservation1);
        voyageRepository.save(voyage1);

        Voyage voyage2 = new Voyage();
        voyage2.setDate(new Date());
        voyage2.setReservation(reservation2);
        voyageRepository.save(voyage2);

        Voyage voyage3 = new Voyage();
        voyage3.setDate(new Date());
        voyage3.setReservation(reservation3);
        voyageRepository.save(voyage3);

        Vol vol1= new Vol();
        vol1.setNom("vol1");
        vol1.setDateDepart(new Date());
        vol1.setDateArrivee(new Date());
        vol1.setVoyage(voyage1);
        volRepository.save(vol1);
        Vol vol2= new Vol();
        vol2.setNom("vol2");
        vol2.setDateDepart(new Date());
        vol2.setDateArrivee(new Date());
        vol2.setVoyage(voyage2);
        volRepository.save(vol2);
        Vol vol3= new Vol();
        vol3.setNom("vol3");
        vol3.setDateDepart(new Date());
        vol3.setDateArrivee(new Date());
        vol3.setVoyage(voyage3);
        volRepository.save(vol3);
        Vol vol4= new Vol();
        vol4.setNom("vol4");
        vol4.setDateDepart(new Date());
        vol4.setDateArrivee(new Date());
        vol4.setVoyage(voyage3);
        volRepository.save(vol4);


        Avion avion1 = new Avion();
        avion1.setNom("avion1");
        avion1.setNbPlace(100);
        avion1.setVol(vol1);
        avionRepository.save(avion1);
        Avion avion2 = new Avion();
        avion2.setNom("avion2");
        avion2.setNbPlace(300);
        avion2.setVol(vol2);
```

```
            avionRepository.save(avion2);
            Avion avion3 = new Avion();
            avion3.setNom("avion3");
            avion3.setNbPlace(200);
            avion3.setVol(vol3);
            avionRepository.save(avion3);


        ;


        };
    }
}
```

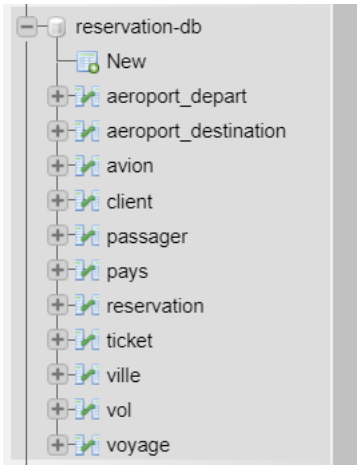


Table Reservation

		date	client_id	id	site_reservation
☐	Edit Copy Delete	2023-05-22	1	1	www.reservation.com
☐	Edit Copy Delete	2023-05-22	2	2	www.reservation.com
☐	Edit Copy Delete	2023-05-22	3	3	www.reservation.com

Table Client

		id	email	nom	prenom
☐	Edit Copy Delete	1	ouassima@gmail.com	khladi	ouassima
☐	Edit Copy Delete	2	mohamed@gmail.com	khladi	mohamed
☐	Edit Copy Delete	3	jinan@gmail.com	akasmiou	jinan

Table Voyage

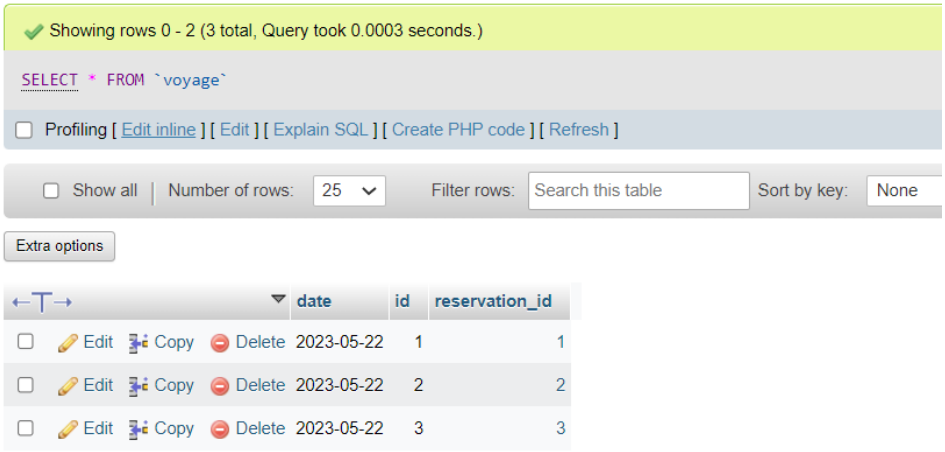


Table Avion

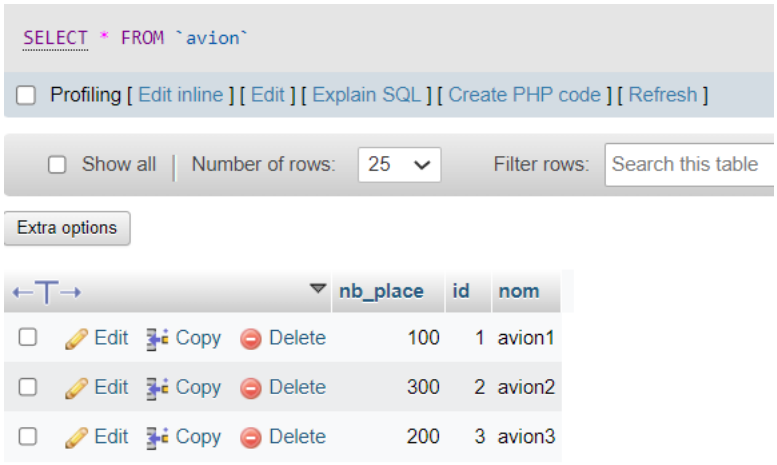


Table Vol

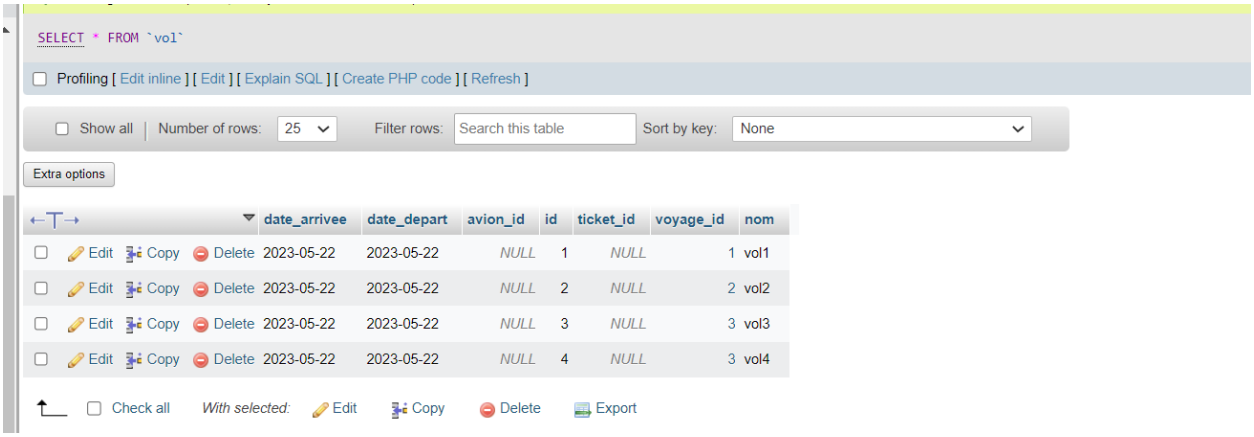


Table Aeroport_Depart

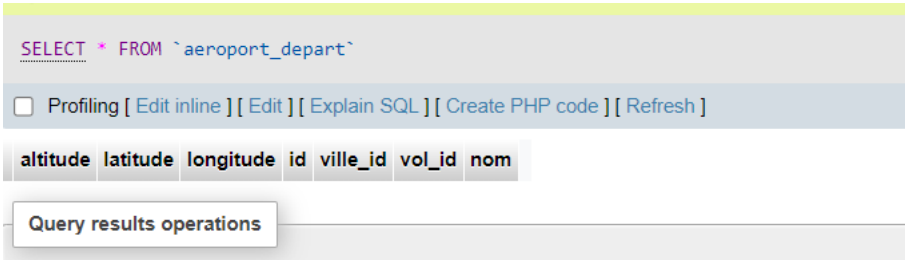


Table Aeroport_Destination

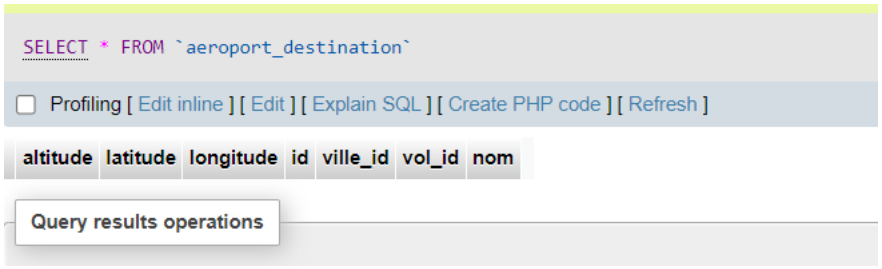


Table Passager

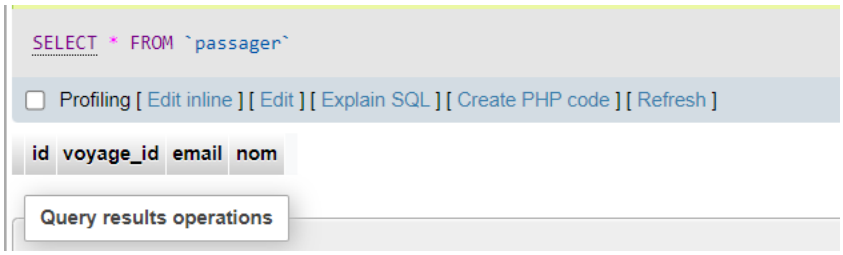


Table Pays

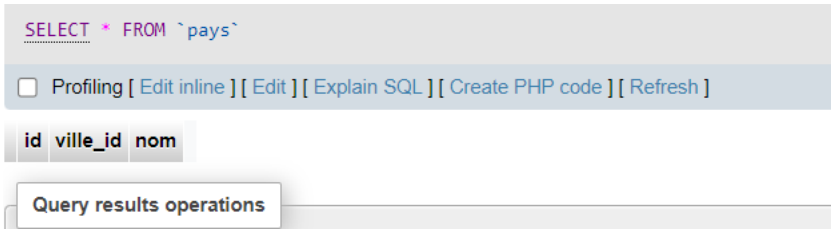
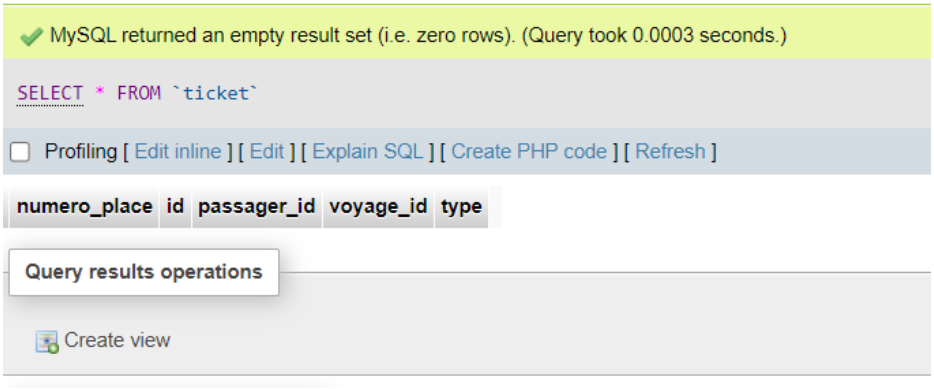


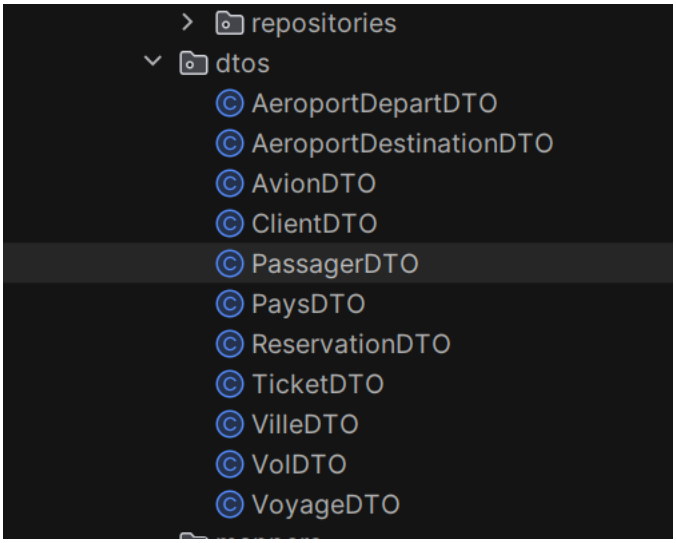
Table Ticket



3. Couche Web REST API :

En créant les DTO et les mappeurs requis :

DTOS :




```java
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.*;
import jakarta.persistence.*;
import lombok.Data;
import org.springframework.format.annotation.DateTimeFormat;

import java.util.Date;

@Data
public class VolDTO {

    private Long id;
    private String nom;
```

```
    private Date dateDepart;

    private Date dateArrivee;
    private AvionDTO avionDTO;
    private VoyageDTO voyageDTO;
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Client;
import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Voyage;
import jakarta.persistence.*;
import lombok.Data;
import org.springframework.format.annotation.DateTimeFormat;

import java.util.Date;
@Data
public class ReservationDTO {

    private Long id;

    private Date date;
    private String siteReservation;

    private ClientDTO clientDTO;

    private VoyageDTO voyageDTO;
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Reservation;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.OneToMany;
import lombok.Data;

import java.util.List;
@Data
public class ClientDTO {


    private Long id;
    private String nom;
    private String prenom;
    private String email;

}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import lombok.Data;
```

```java
@Data
public class PaysDTO {
    private Long id;
    private String nom;
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import akasmiou.ouassima.akasmiou_ouassima.dao.enums.Type;
import jakarta.persistence.EnumType;
import jakarta.persistence.Enumerated;
import lombok.Data;

@Data
public class TicketDTO {
    private Long id;
    private int numeroPlace;

    private Type type;
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import lombok.Data;

@Data
public class VilleDTO {
    private Long id;
    private String nom;
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import jakarta.persistence.Temporal;
import jakarta.persistence.TemporalType;
import lombok.Data;
import org.springframework.format.annotation.DateTimeFormat;

import java.util.Date;
@Data
public class VoyageDTO {
    private Long id;

    private Date date;
}
```

```java
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import lombok.Data;

@Data
public class AvionDTO {
```

```
    private Long id;
    private String nom;
    private int nbPlace;
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import lombok.Data;

@Data
public class AeroportDepartDTO {
    private Long id;
    private String nom;
    private float longitude;
    private float latitude;
    private float altitude;
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import lombok.Data;

@Data
public class AeroportDestinationDTO {
    private Long id;
    private String nom;
    private float longitude;
    private float latitude;
    private float altitude;
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.dtos;

import lombok.Data;

@Data
public class PassagerDTO {
    private Long id;
    private String nom;
    private String email;
}
```

Mappeurs :

```java
package akasmiou.ouassima.akasmiou_ouassima.mappers;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.*;
import akasmiou.ouassima.akasmiou_ouassima.dtos.*;
import org.springframework.beans.BeanUtils;
import org.springframework.stereotype.Service;

@Service
public class ReservationMapperImpl {
    public VolDTO fromVol(Vol vol) {
        VolDTO volDTO = new VolDTO();
        BeanUtils.copyProperties(vol, volDTO);
        volDTO.setAvionDTO(fromAvion(vol.getAvion()));
        volDTO.setVoyageDTO(fromVoyage(vol.getVoyage()));
        return volDTO;
    }
    public Vol fromVolDTO(VolDTO volDTO) {
        Vol  vol= new Vol();
        BeanUtils.copyProperties(volDTO, vol);
        return vol;
    }
    public ReservationDTO fromReservation(Reservation reservation) {
        ReservationDTO reservationDTO = new ReservationDTO();
        BeanUtils.copyProperties(reservation, reservationDTO);
        reservationDTO.setClientDTO(fromClient(reservation.getClient()));
        reservationDTO.setVoyageDTO(fromVoyage(reservation.getVoyage()));
        return reservationDTO;
    }
    public Reservation fromReservationDTO(ReservationDTO reservationDTO) {
        Reservation reservation = new Reservation();
        BeanUtils.copyProperties(reservationDTO, reservation);
        return reservation;
    }

    public AvionDTO fromAvion(Avion avion) {
        AvionDTO avionDTO = new AvionDTO();
        BeanUtils.copyProperties(avion, avionDTO);
        return avionDTO;
    }
    public Avion fromAvionDTO(AvionDTO avionDTO) {
        Avion avion= new Avion();
        BeanUtils.copyProperties(avionDTO, avion);
        return avion;
    }
    public VoyageDTO fromVoyage(Voyage voyage) {
        VoyageDTO voyageDTO = new VoyageDTO();
        BeanUtils.copyProperties(voyage, voyageDTO);
        return voyageDTO;
    }
    public Voyage fromVoyageDTO(VoyageDTO voyageDTO) {
        Voyage voyage = new Voyage();
        BeanUtils.copyProperties(voyageDTO, voyage);
        return voyage;
    }
    public ClientDTO fromClient(Client client) {
        ClientDTO clientDTO = new ClientDTO();
```

```
        BeanUtils.copyProperties(client, clientDTO);
        return clientDTO;
    }
    public Client fromClientDTO(ClientDTO clientDTO) {
        Client client = new Client();
        BeanUtils.copyProperties(clientDTO, client);
        return client;
    }
}
```

```
package akasmiou.ouassima.akasmiou_ouassima.services;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Vol;
import akasmiou.ouassima.akasmiou_ouassima.dtos.ReservationDTO;
import akasmiou.ouassima.akasmiou_ouassima.dtos.VolDTO;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public interface ReservationService {
    VolDTO saveVol(VolDTO volDTO);
    List<VolDTO> listVols();
    VolDTO getVol(Long volId);
    VolDTO updateVol(VolDTO volDTO);
    void deleteVol(Long volId);
    ReservationDTO saveReservation(ReservationDTO reservationDTO);
    List<ReservationDTO> listReservations();
    ReservationDTO getReservation(Long reservationId);
    ReservationDTO updateReservation(ReservationDTO reservationDTO);
    void deleteReservation(Long reservationId);

}
```

```
package akasmiou.ouassima.akasmiou_ouassima.services;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Reservation;
import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Vol;
import
akasmiou.ouassima.akasmiou_ouassima.dao.repositories.ReservationRepository;
import akasmiou.ouassima.akasmiou_ouassima.dao.repositories.VolRepository;
import akasmiou.ouassima.akasmiou_ouassima.dtos.ReservationDTO;
import akasmiou.ouassima.akasmiou_ouassima.dtos.VolDTO;
import akasmiou.ouassima.akasmiou_ouassima.mappers.ReservationMapperImpl;

import java.util.List;
import java.util.stream.Collectors;

@Service
@Transactional
@AllArgsConstructor
public class ReservationServiceImpl implements ReservationService {
    VolRepository volRepository;
    ReservationRepository reservationRepository;
    ReservationMapperImpl dtoMapper;
```

```
    @Override
    public VolDTO saveVol(VolDTO volDTO) {
        Vol vol = dtoMapper.fromVolDTO(volDTO);
        Vol savedVol = volRepository.save(vol);
        return dtoMapper.fromVol(savedVol);
    }

    @Override
    public List<VolDTO> listVols() {

        List<Vol> vols = volRepository.findAll();
        List<VolDTO> collect =vols.stream().map(vol ->
dtoMapper.fromVol(vol).collect(Collectors.toList());
        return collect;
        return null;
    }

    @Override
    public VolDTO getVol(Long volId) {
        Vol vol = volRepository.findById(volId).orElse(null);
        return dtoMapper.fromVol(vol);
    }

    @Override
    public VolDTO updateVol(VolDTO volDTO) {

        Vol vol = dtoMapper.fromVolDTO(volDTO);
        Vol savedVol = volRepository.save(vol);
        return dtoMapper.fromVol(savedVol);

    }

    @Override
    public void deleteVol(Long volId) {
         volRepository.deleteById(volId);
    }

    @Override
    public ReservationDTO saveReservation(ReservationDTO reservationDTO) {
        Reservation reservation =
dtoMapper.fromReservationDTO(reservationDTO);
        Reservation savedReservation =
reservationRepository.save(reservation);
        return dtoMapper.fromReservation(savedReservation);
    }

    @Override
    public List<ReservationDTO> listReservations() {
        List<ReservationDTO> collect =
reservationRepository.findAll().stream().map(reservation ->
dtoMapper.fromReservation(reservation)).collect(Collectors.toList());
        return collect;

    }

    @Override
    public ReservationDTO getReservation(Long reservationId) {
        Reservation reservation =
```

```
reservationRepository.findById(reservationId).orElse(null);
        return dtoMapper.fromReservation(reservation);

    }

    @Override
    public ReservationDTO updateReservation(ReservationDTO reservationDTO) {
        Reservation reservation =
dtoMapper.fromReservationDTO(reservationDTO);
        Reservation savedReservation =
reservationRepository.save(reservation);
        return dtoMapper.fromReservation(savedReservation);
    }

    @Override
    public void deleteReservation(Long reservationId) {
          reservationRepository.deleteById(reservationId);
    }
}
```

- Créer le Web service RESTful qui permet de gérer les Vols

VolController :

```
package akasmiou.ouassima.akasmiou_ouassima.web;

import akasmiou.ouassima.akasmiou_ouassima.dao.entities.Vol;
import akasmiou.ouassima.akasmiou_ouassima.dtos.VolDTO;
import akasmiou.ouassima.akasmiou_ouassima.services.ReservationService;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
public class VolController {
      private ReservationService reservationService;

    public VolController(ReservationService reservationService) {
        this.reservationService = reservationService;
    }

    @GetMapping("/vols")
    public List<VolDTO> vols(){
        return reservationService.listVols();
    }

    @GetMapping("/vols/{id}")
    public VolDTO getVol(@PathVariable(name = "id") Long volId){
        return reservationService.getVol(volId);
    }
    @PostMapping("vols")
    public VolDTO saveVol(@RequestBody VolDTO volDTO){
        return reservationService.saveVol(volDTO);
    }
    @PutMapping("/vols/{volId}")
    public VolDTO updateVol(@PathVariable Long volId, @RequestBody VolDTO
volDTO){
        volDTO.setId(volId);
        return reservationService.updateVol(volDTO);
```

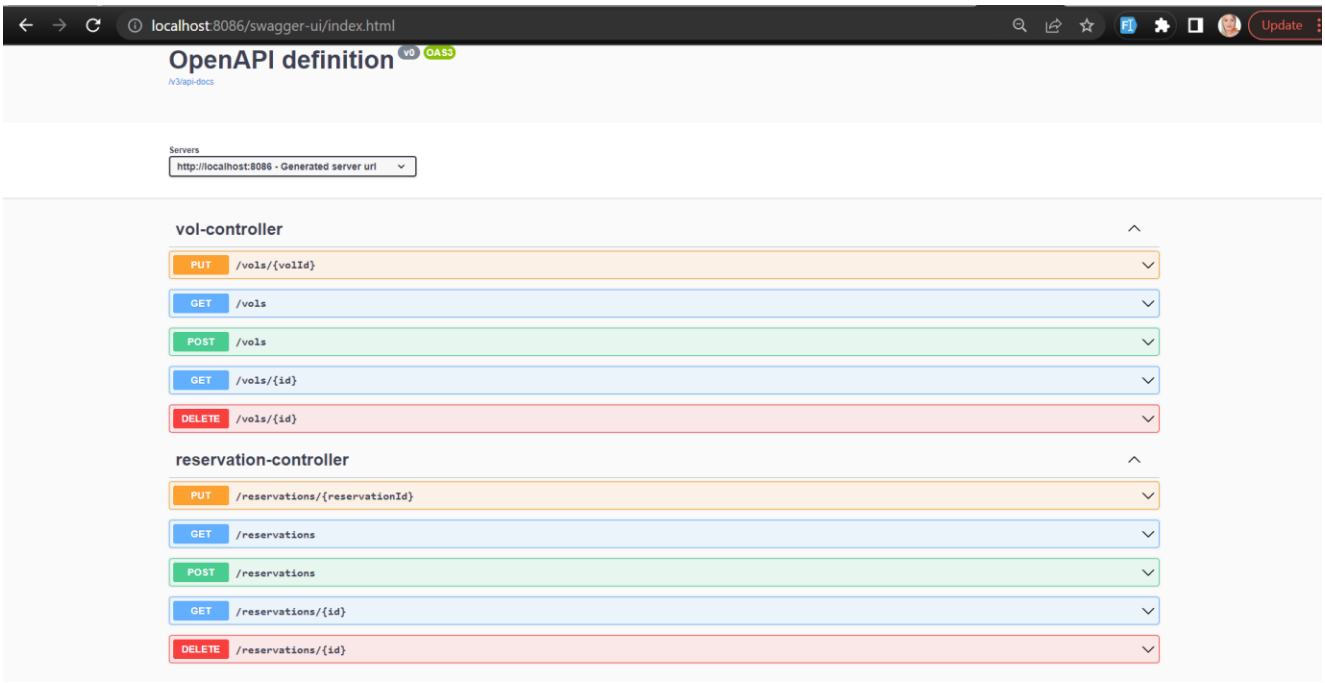
```
    }
    @DeleteMapping("/vols/{id}")
    public void deleteVol(@PathVariable Long id){
        reservationService.deleteVol(id);
    }

}
```

- Créer le Web service RESTful qui permet de gérer les réservations

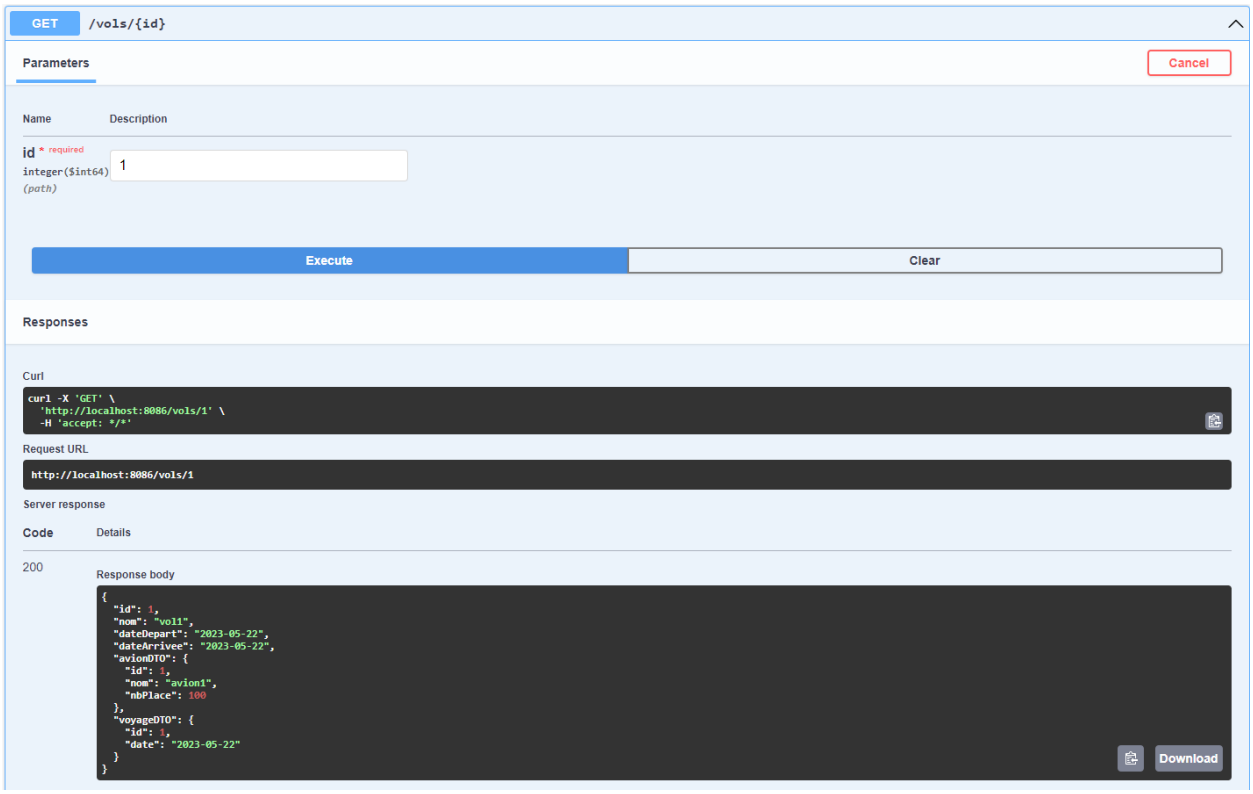
ReservationController :

```
package akasmiou.ouassima.akasmiou_ouassima.web;

import akasmiou.ouassima.akasmiou ouassima.dtos.ReservationDTO;
import akasmiou.ouassima.akasmiou_ouassima.services.ReservationService;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
public class ReservationController {

    private ReservationService reservationService;

    public ReservationController(ReservationService reservationService) {
        this.reservationService = reservationService;
    }



    @GetMapping("/reservations")
    public List<ReservationDTO> reservations(){
        return reservationService.listReservations();
    }

    @GetMapping("/reservations/{id}")
    public ReservationDTO getReservation(@PathVariable(name = "id") Long
reservationId){
        return reservationService.getReservation(reservationId);
    }
    @PostMapping("reservations")
    public ReservationDTO saveReservation(@RequestBody ReservationDTO
reservationDTO){
        return reservationService.saveReservation(reservationDTO);
    }
    @PutMapping("/reservations/{reservationId}")
    public ReservationDTO updateReservation(@PathVariable Long reservationId,
@RequestBody ReservationDTO reservationDTO){
        reservationDTO.setId(reservationId);
        return reservationService.updateReservation(reservationDTO);
    }
    @DeleteMapping("/reservations/{id}")
    public void deleteReservation(@PathVariable Long id){
        reservationService.deleteReservation(id);
    }
}
```

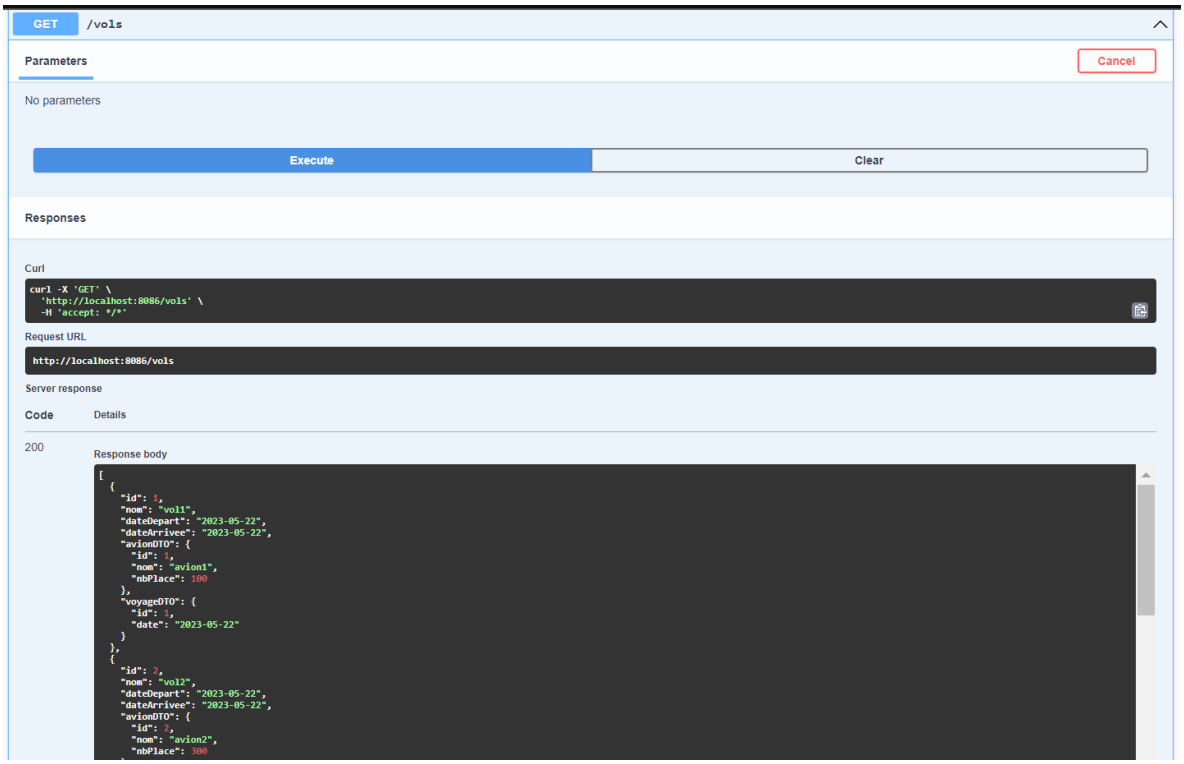
- Générer la documentation SWAGGER des API RESTful



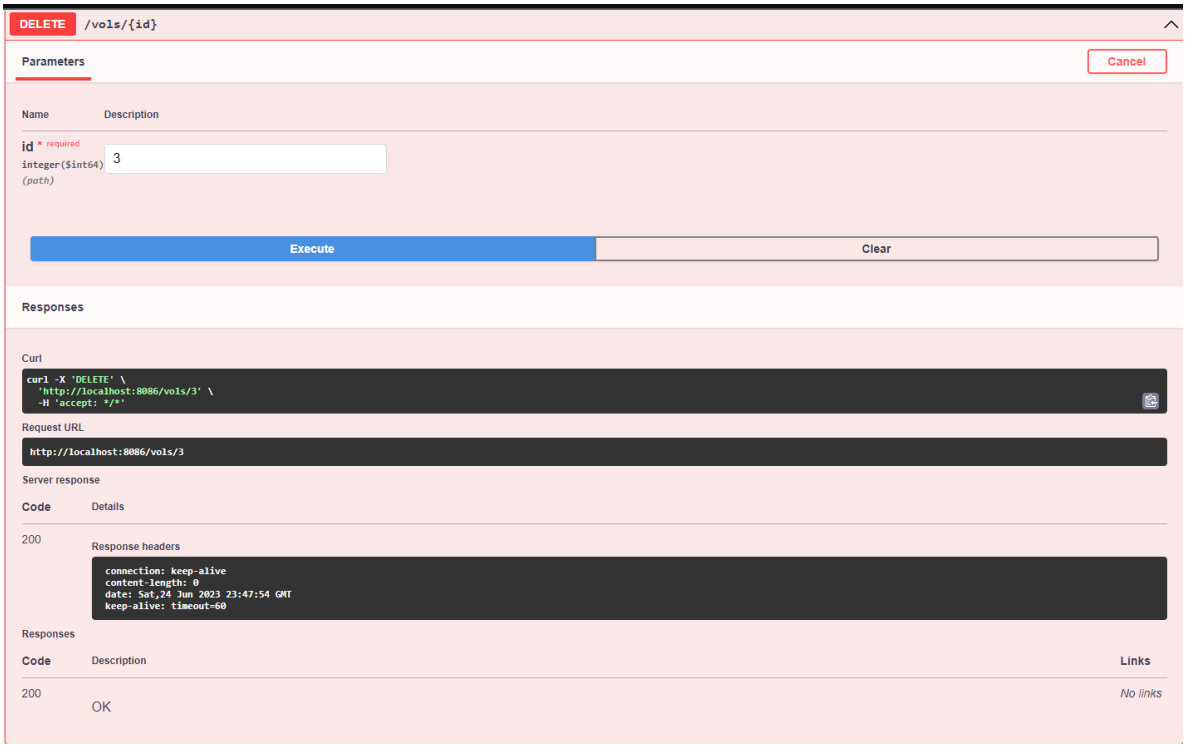
Méthode Get vol By Id :



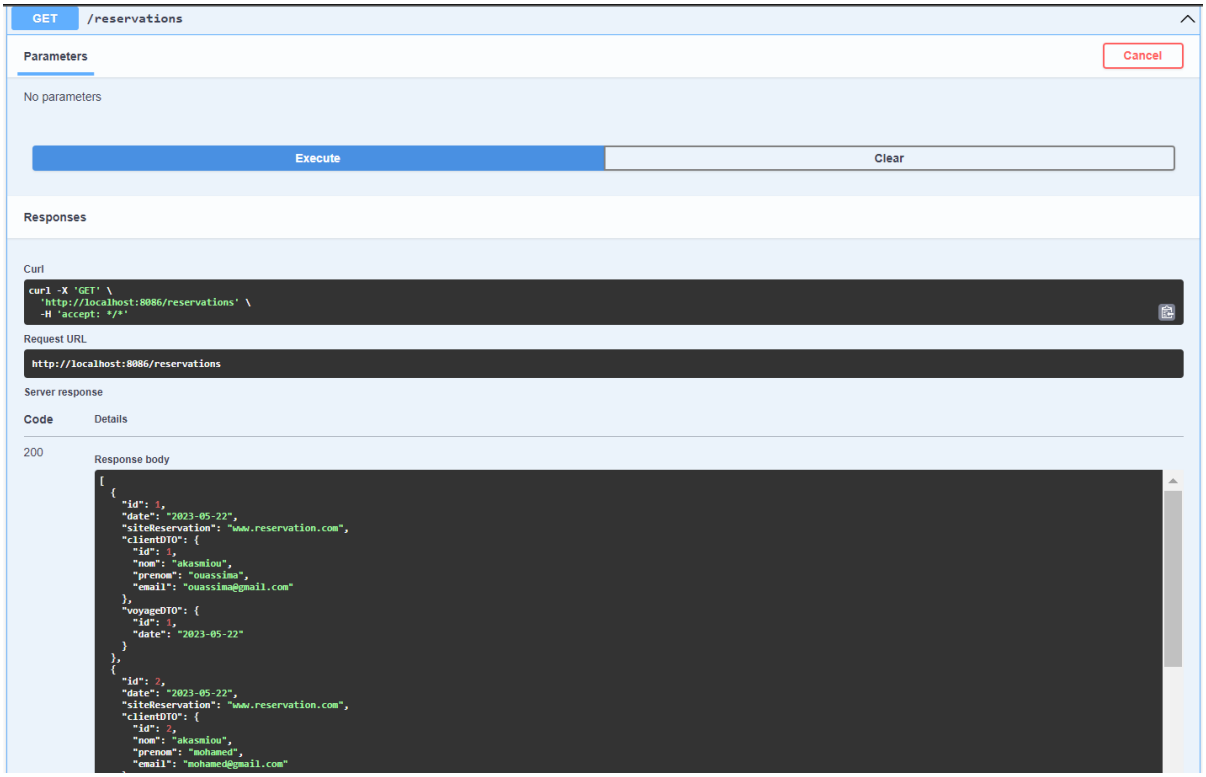
Méthode Get pour récupérer l'ensemble des vols :



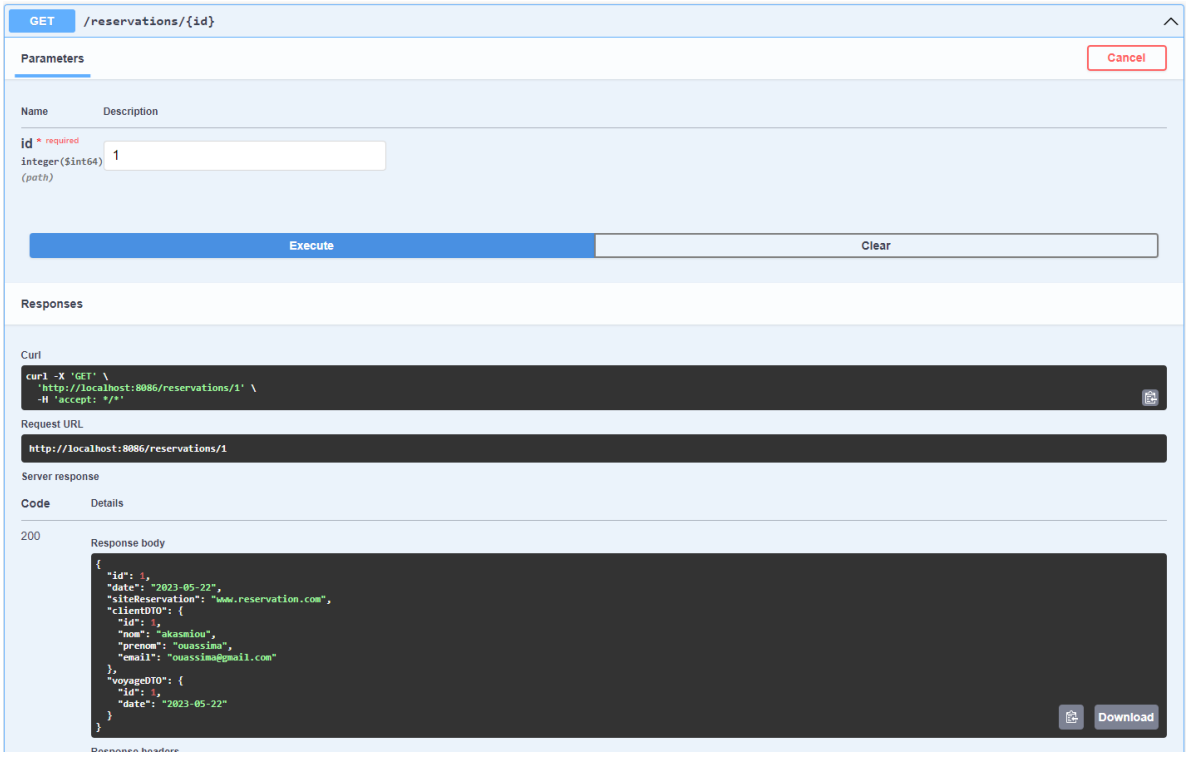
Méthode Delete vol:



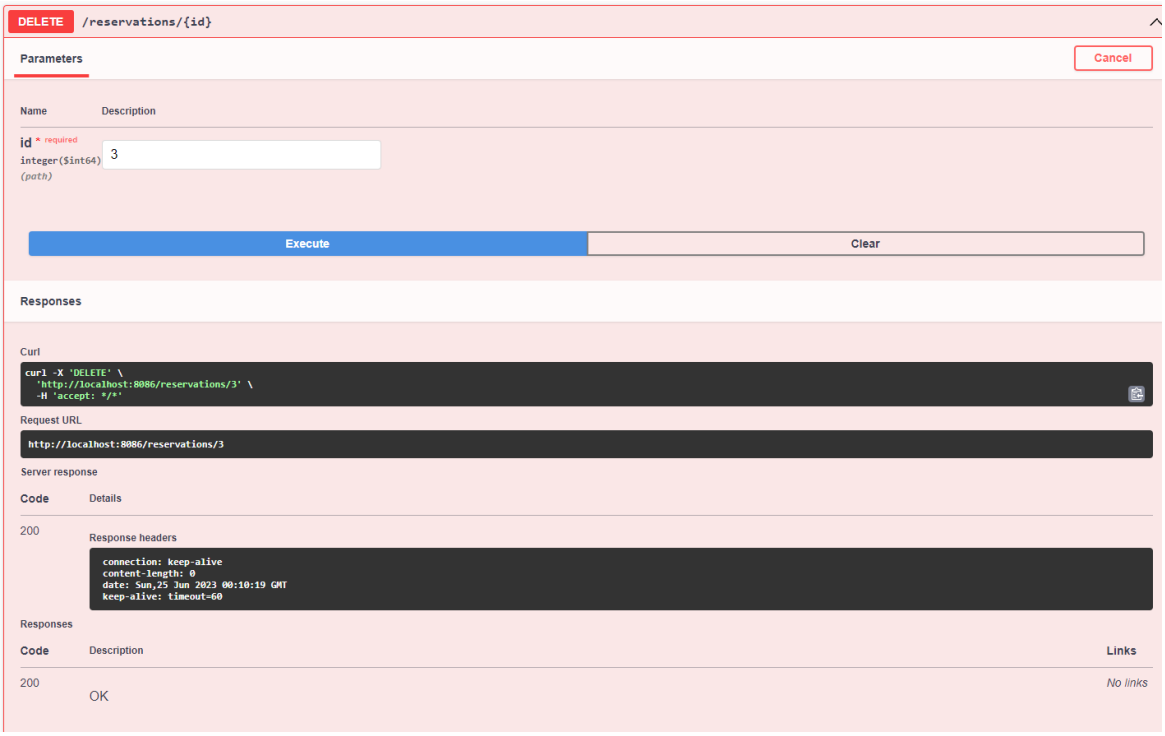
Méthode Get pour récupérer l'ensemble des réservations :



Méthode Get réservation By Id :

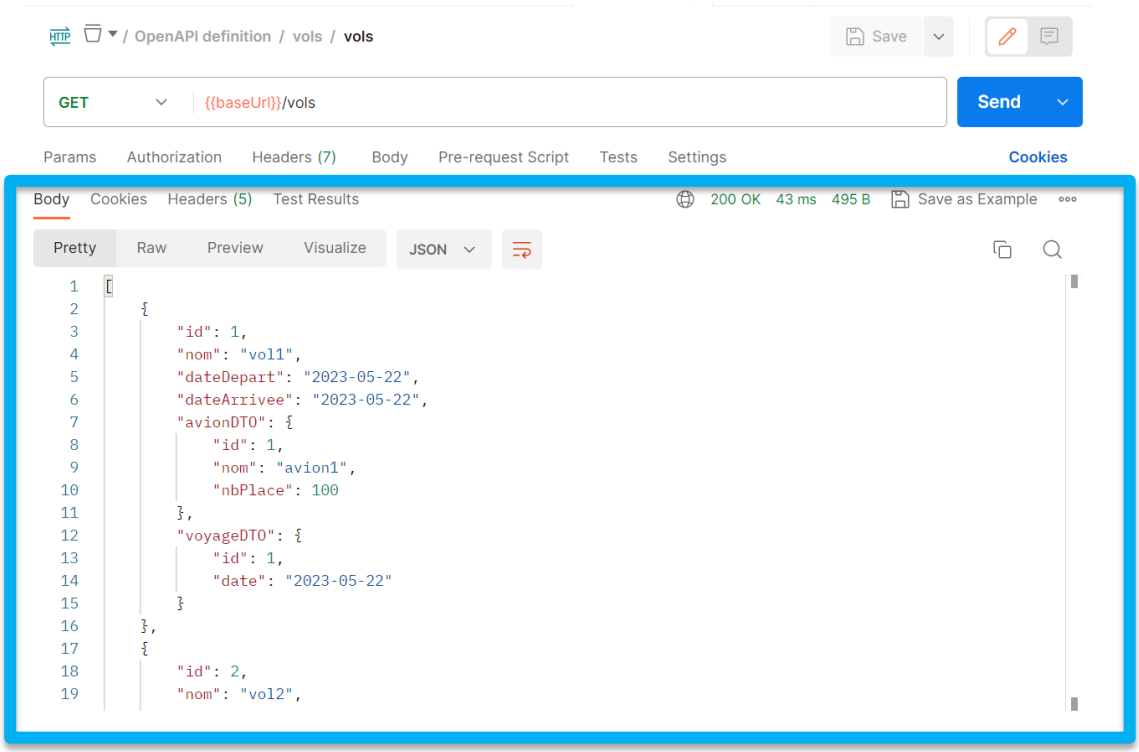


Méthode Delete réservation:

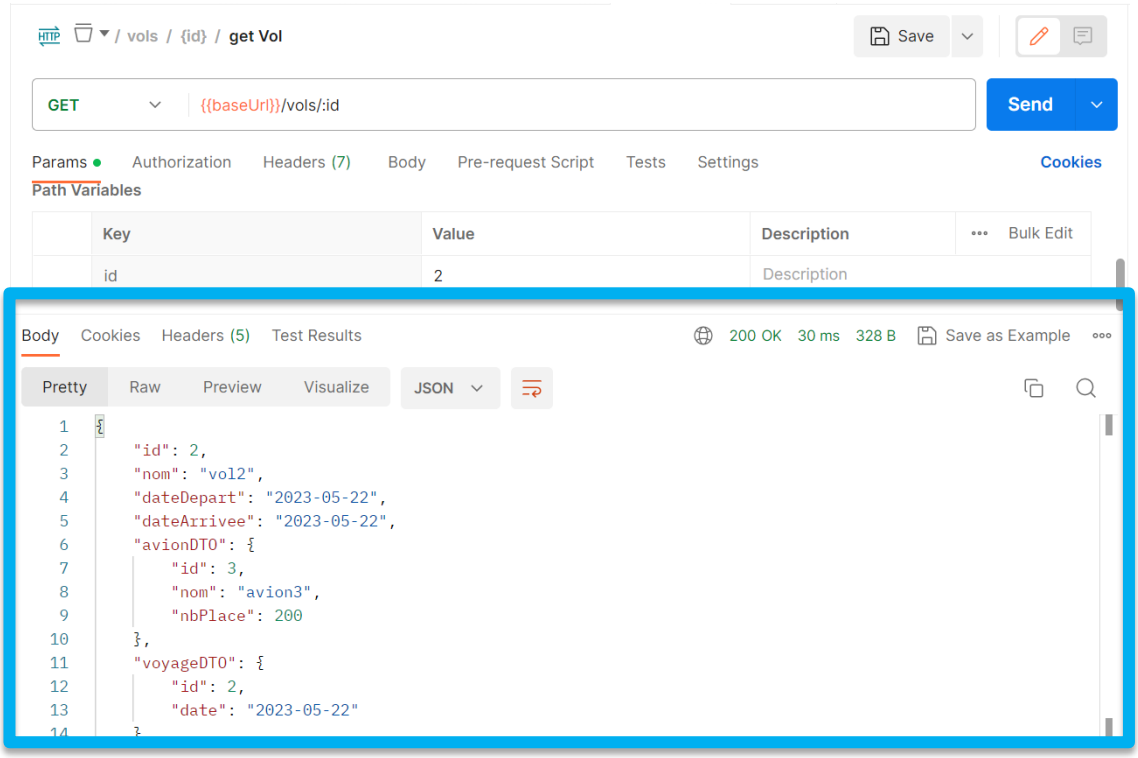


- Tester les Web service avec un client REST comme Postman

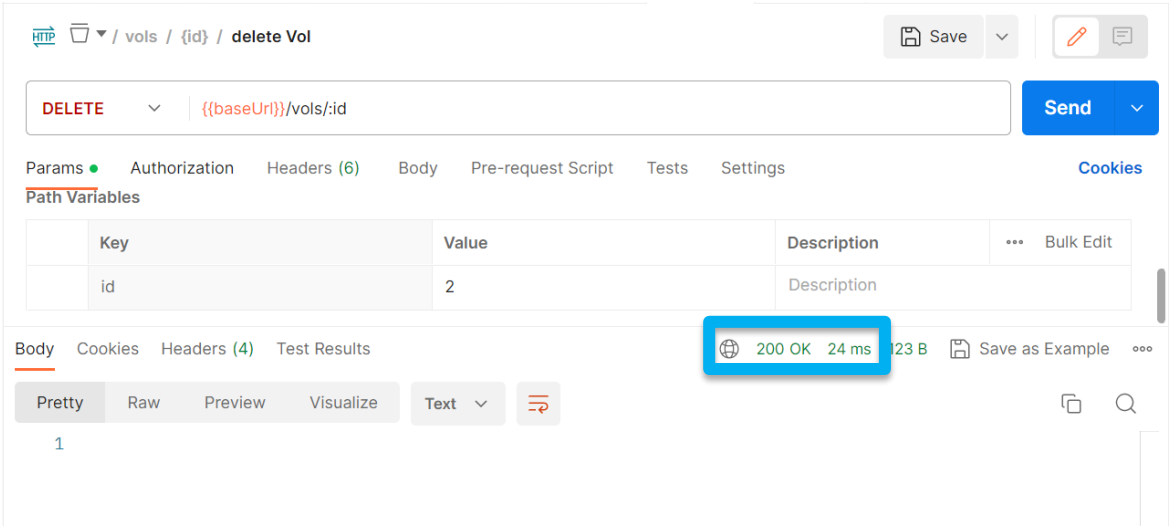
Méthode Get (récupérer l'ensemble des vols) :



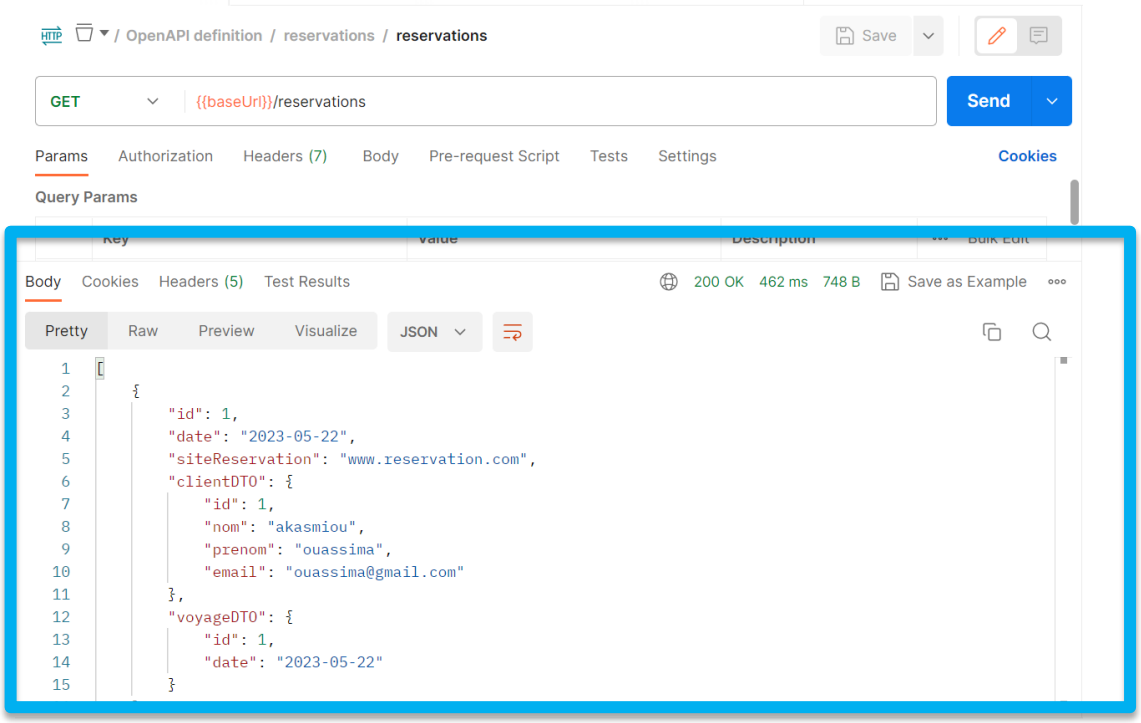
Méthode Get vol By Id :



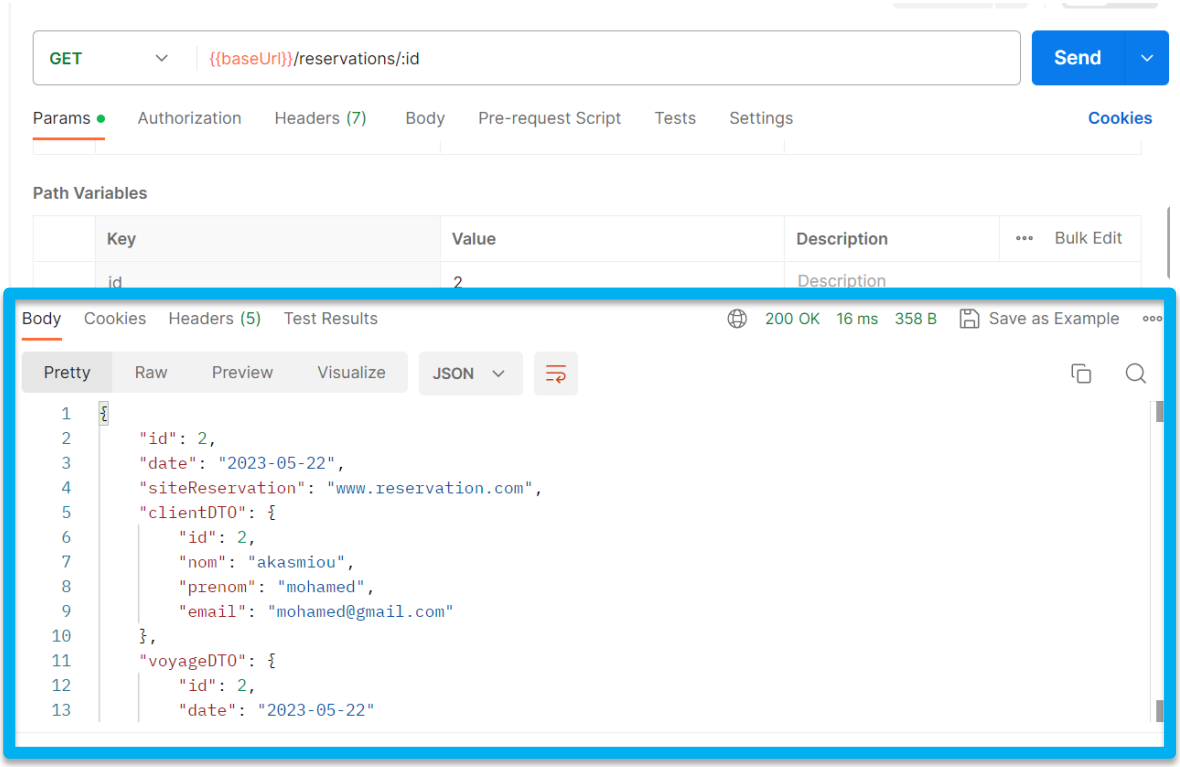
Méthode Delete vol:



Méthode Get pour récupérer l'ensemble des réservations :



Méthode Get réservation By Id :



Méthode Delete réservation: