

# Flight Reservation System (Monolithic - In-Memory)

---

## 1. application.properties

---

```
spring.datasource.url=jdbc:h2:mem:flightdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
springdoc.swagger-ui.path=/swagger-ui.html
```

## 2. Flight.java

---

```
@Entity
public class Flight {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(unique = true)
    private String flightNumber;
    private String origin;
    private String destination;
    private LocalDateTime departureTime;
    private int seatsAvailable;

    @OneToMany(mappedBy = "flight", cascade = CascadeType.ALL)
    private List<Reservation> reservations;

    // Getters and Setters
}
```

## 3. Reservation.java

---

```
@Entity
public class Reservation {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String passengerName;
    private String passengerEmail;
    private int seatsBooked;
    private LocalDateTime reservedAt;

    @ManyToOne
    @JoinColumn(name = "flight_id")
    private Flight flight;

    // Getters and Setters
}
```

```
}
```

#### 4. FlightRepository.java

-----

```
public interface FlightRepository extends JpaRepository<Flight, Long> {  
    boolean existsByFlightNumber(String flightNumber);  
}
```

#### 5. ReservationRepository.java

-----

```
public interface ReservationRepository extends JpaRepository<Reservation, Long> {  
    List<Reservation> findByFlightId(Long flightId);  
}
```

#### 6. FlightService.java

-----

```
@Service  
public class FlightService {  
    @Autowired private FlightRepository flightRepository;  
  
    public Flight addFlight(Flight flight) { ... }  
    public List<Flight> getAllFlights() { ... }  
    public Flight getFlightById(Long id) { ... }  
    public Flight updateFlight(Long id, Flight updatedFlight) { ... }  
    public void deleteFlight(Long id) { ... }  
}
```

#### 7. ReservationService.java

-----

```
@Service  
public class ReservationService {  
    @Autowired private ReservationRepository reservationRepository;  
    @Autowired private FlightRepository flightRepository;  
  
    public Reservation bookReservation(Long flightId, Reservation reservation) { ... }  
    public List<Reservation> getAllReservations() { ... }  
    public List<Reservation> getReservationsByFlight(Long flightId) { ... }  
    public void cancelReservation(Long reservationId) { ... }  
}
```

#### 8. FlightController.java

-----

```
@RestController  
@RequestMapping("/api/flights")  
public class FlightController {  
    @Autowired private FlightService flightService;  
  
    @PostMapping public ResponseEntity<Flight> addFlight(@RequestBody Flight flight) { ... }  
    @GetMapping public ResponseEntity<List<Flight>> getAllFlights() { ... }  
    @GetMapping("/{id}") public ResponseEntity<Flight> getFlightById(@PathVariable Long id) { ... }  
}
```

```
@PutMapping("/{id}") public ResponseEntity<Flight> updateFlight(...) { ... }  
@DeleteMapping("/{id}") public ResponseEntity<Void> deleteFlight(...) { ... }  
}
```

## 9. ReservationController.java

```
-----  
@RestController  
@RequestMapping("/api/reservations")  
public class ReservationController {  
    @Autowired private ReservationService reservationService;  
  
    @PostMapping public ResponseEntity<Reservation> bookReservation(...) { ... }  
    @GetMapping public ResponseEntity<List<Reservation>> getAllReservations() { ... }  
    @GetMapping("/flight/{flightId}") public ResponseEntity<List<Reservation>> getByFlight(...) { ... }  
}  
    @DeleteMapping("/{id}") public ResponseEntity<Void> cancelReservation(...) { ... }  
}
```

## 10. Swagger UI Link

```
-----  
http://localhost:8080/swagger-ui.html
```