### CRUD JUnit Tests for Spring Data JPA - Testing Certification Repository Layer

# @DataJpaTest Annotation :

The Spring boot provides @DataJpaTest annotation. This annotation will disable full auto-configuration and instead apply only configuration relevant to JPA tests.

The @DataJpaTest annotation doesn’t load other Spring beans (@Component, @Controller, @Service, and annotated beans) into ApplicationContext.

# 1.Create Spring Boot Project

**Project Name:** MyCabBookingProject

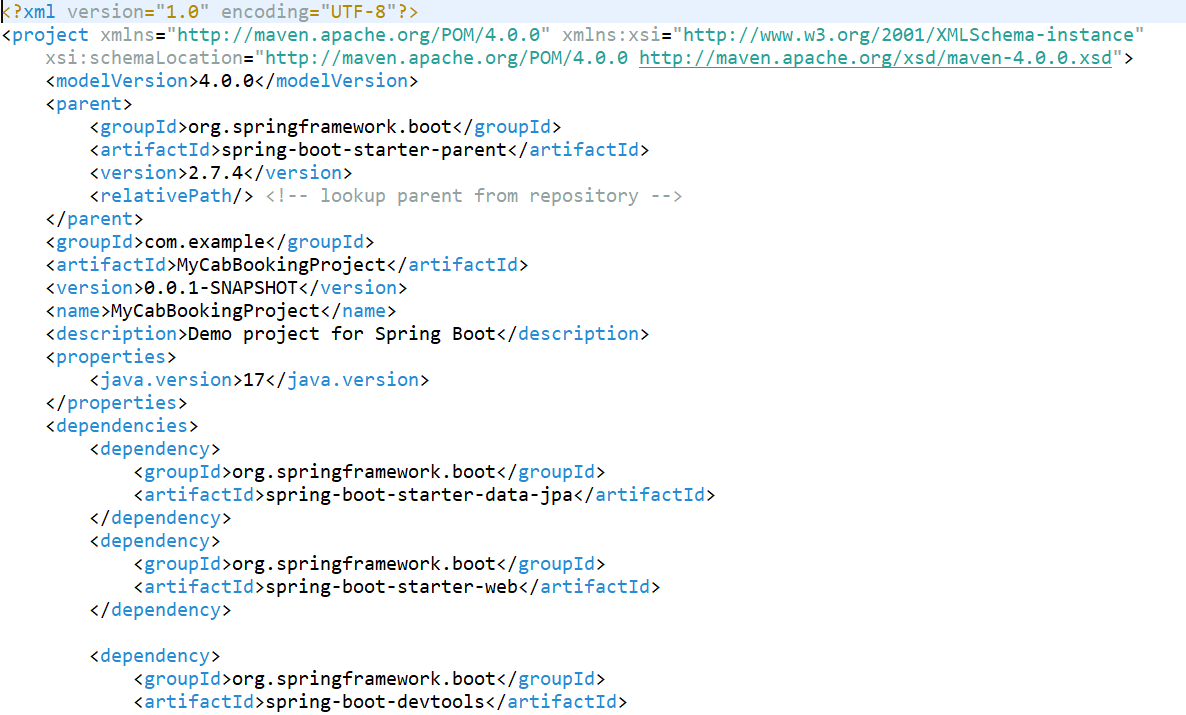
**Project Type:**Maven

**Choose dependencies:**  Spring Data JPA, MYSQL database, Spring Boot Dev Tools

**Package name:**com.cabbook

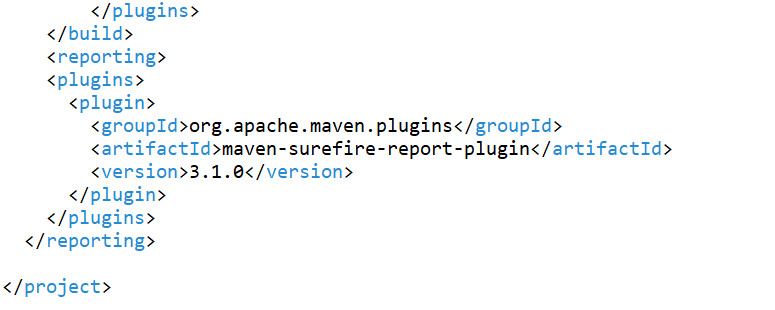
# 2. Maven Dependencies

Pom.xml :





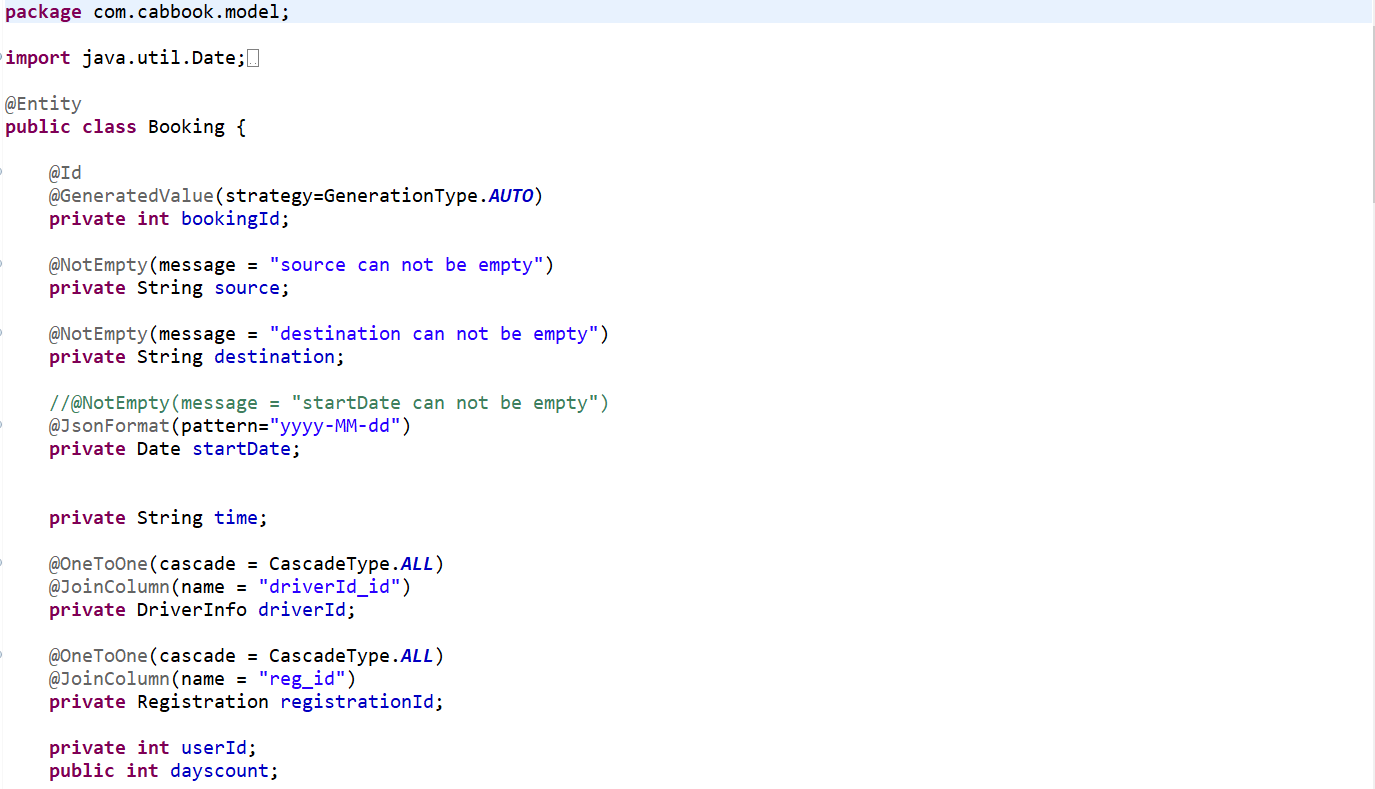




3. Create JPA Entity

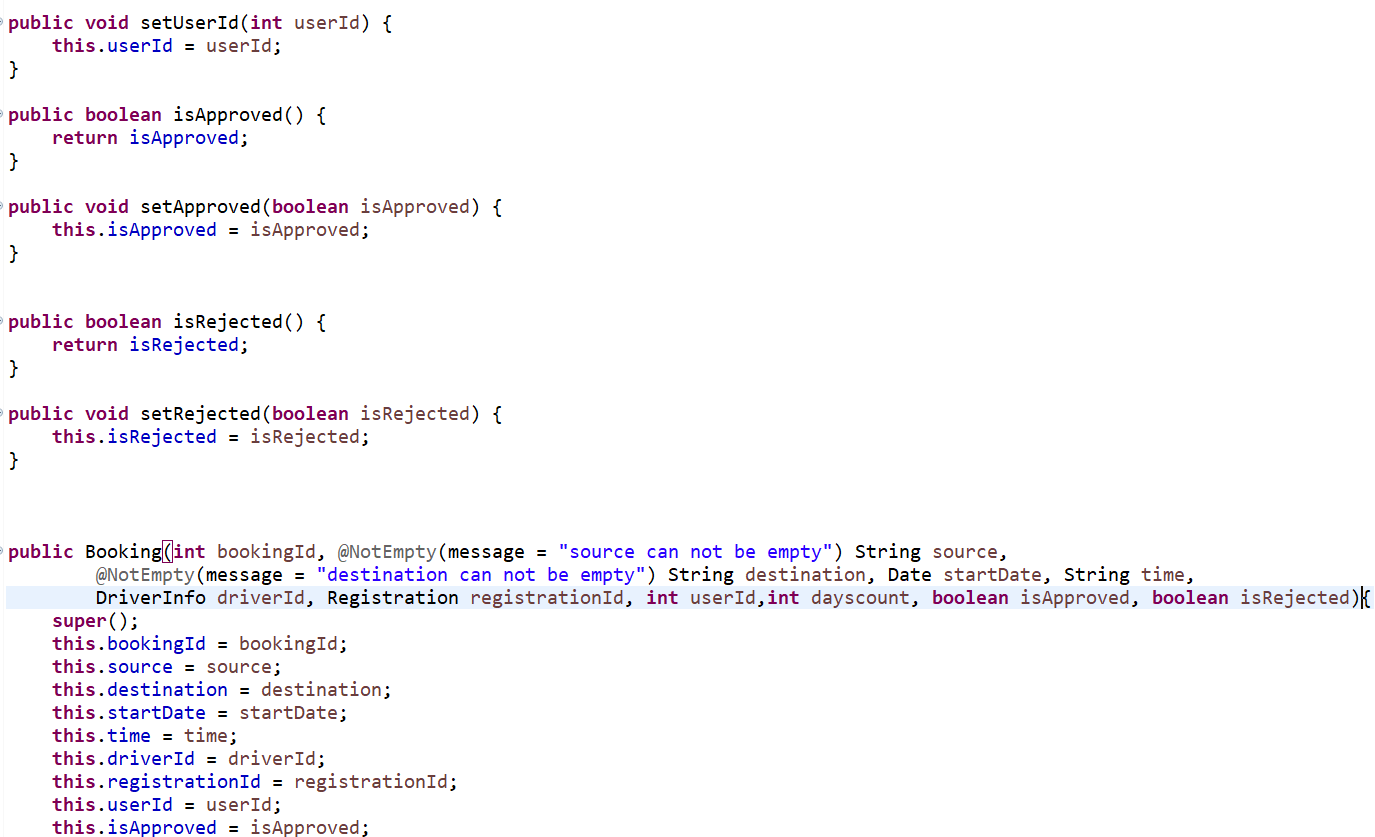
Let's create an entity package inside a base package "com.dxc.fooddelivery.entity". Within the entity package, create a Restaurant and FoodItem classes.

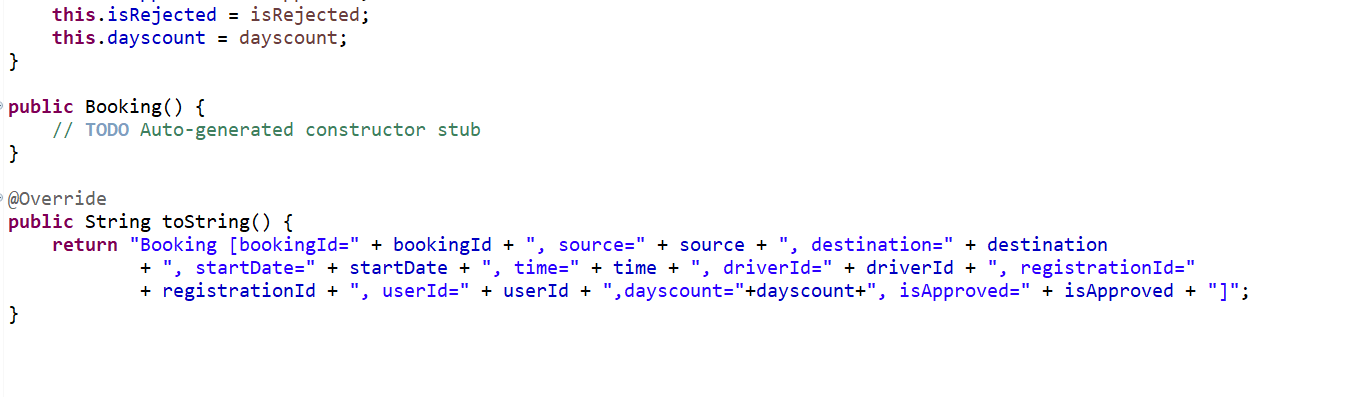
**Booking.java**



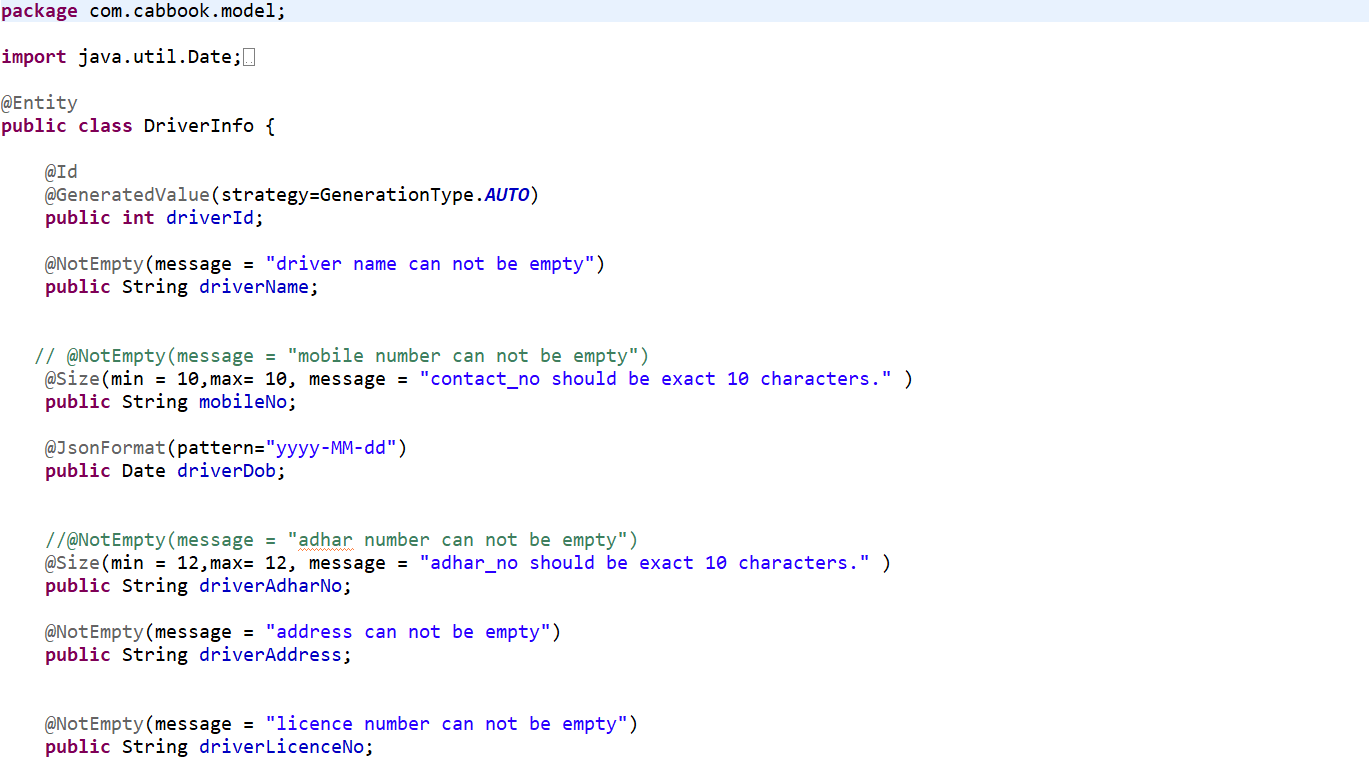
****

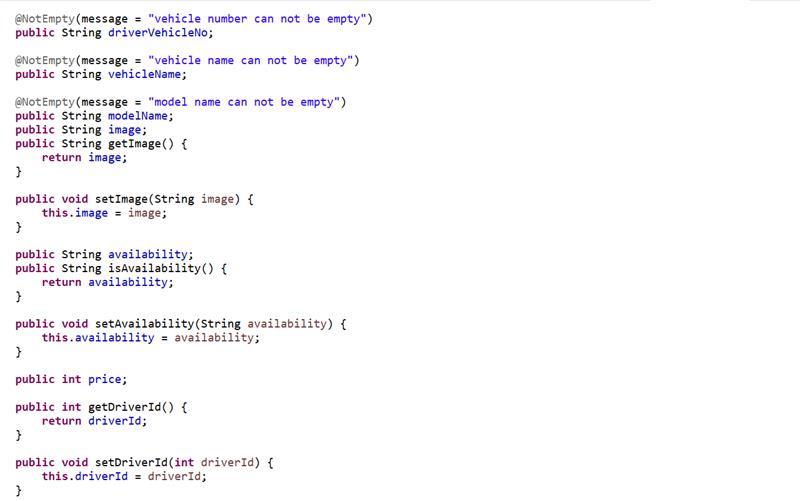
**DriverInfo.java**

****



**DriverInfo.java:**

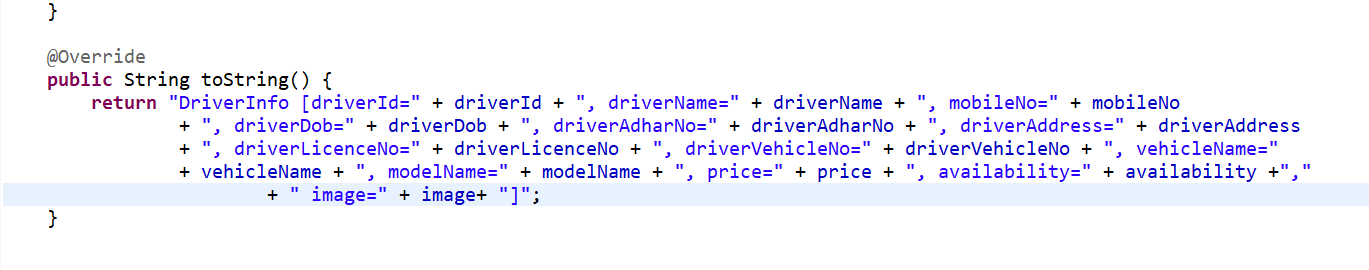
****

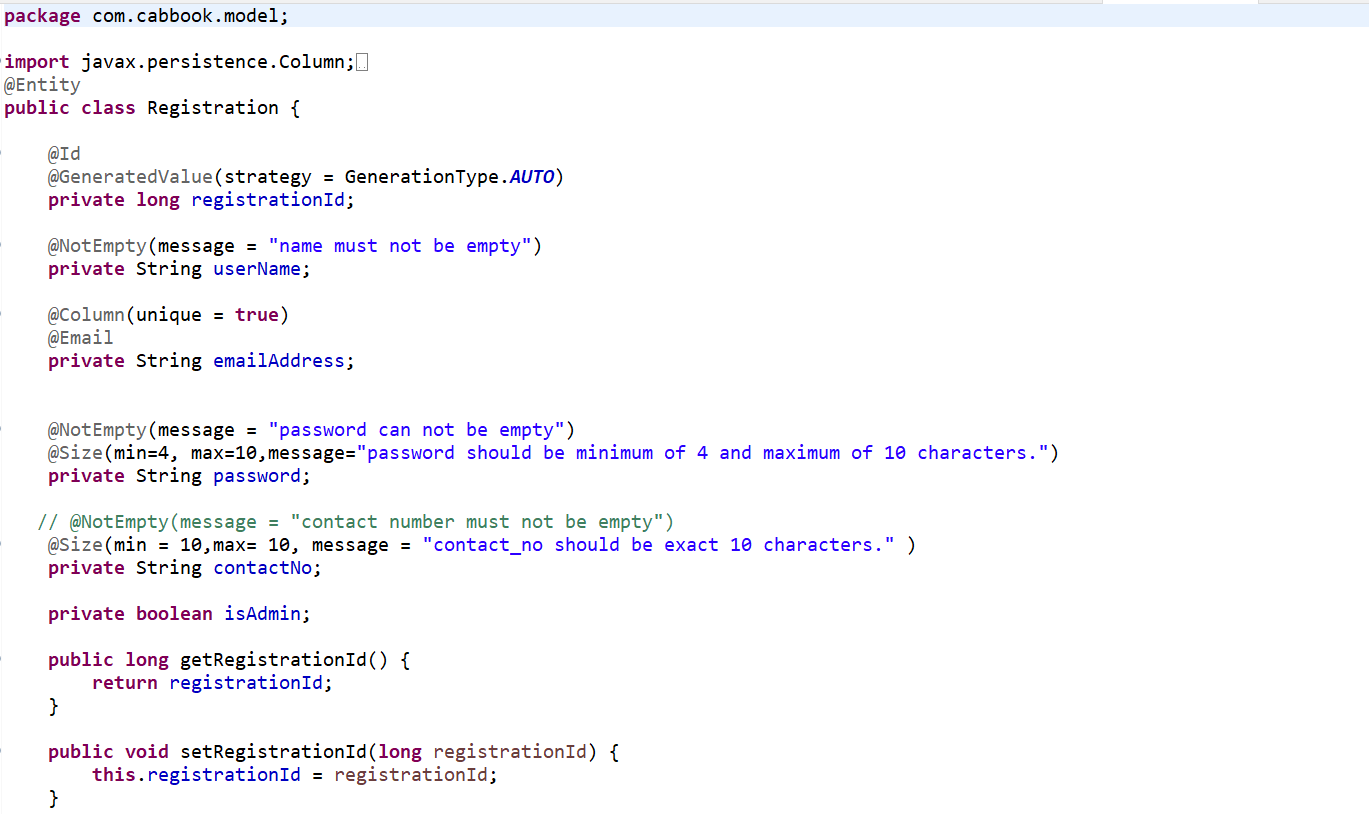
****

# 

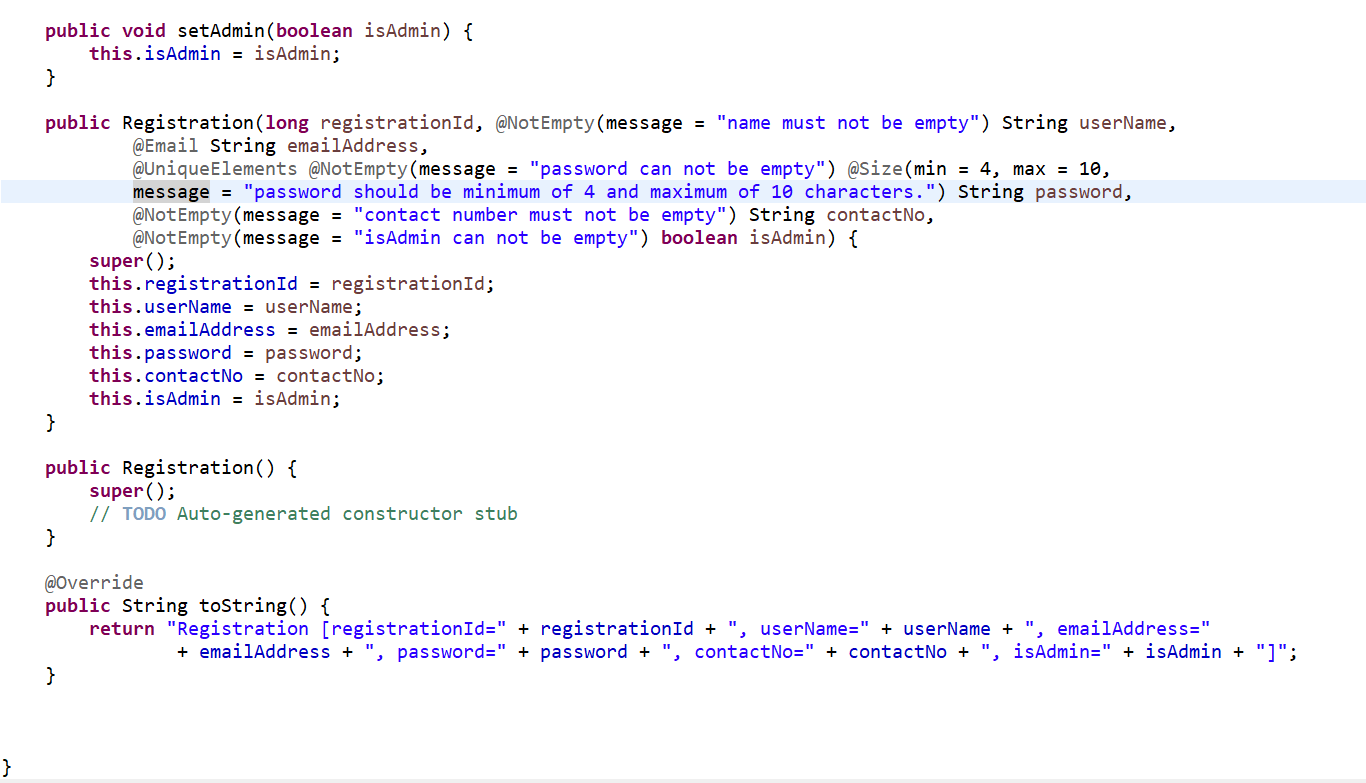
# 

# 



**Registration.java:**

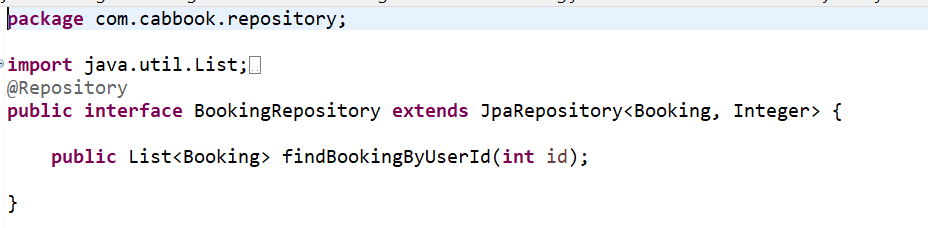
# 



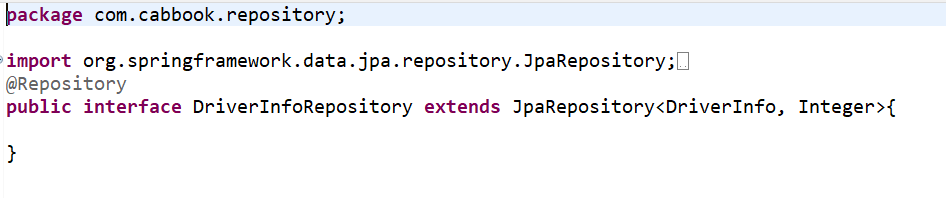
# 4. Create Spring Data JPA Repository

Let's create a repository package inside a base package "com.dxc.fooddelivery.repository". Within the repository package, create a RestaurantRepo and FoodItemRepo interfaces with the following content:

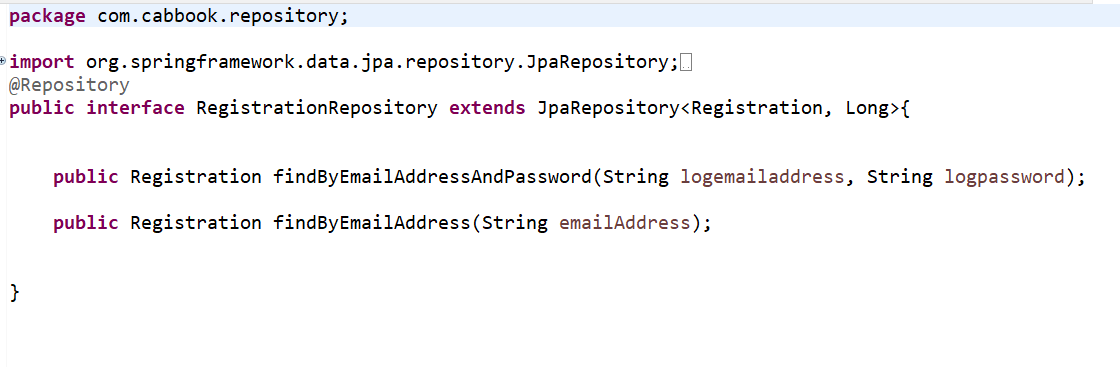
**BookingRepository.java**



**DriverInfoRepository.java**



**RegistrationRepository.java**

****

# 5. JUnit Tests for Spring Data JPA Repository:

Create a MyCabBookingServiceTest class with the following content:



Test Report Result :

