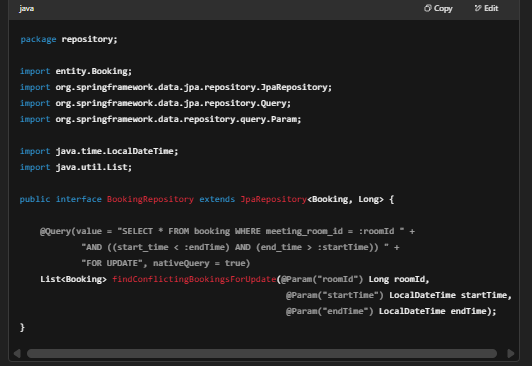
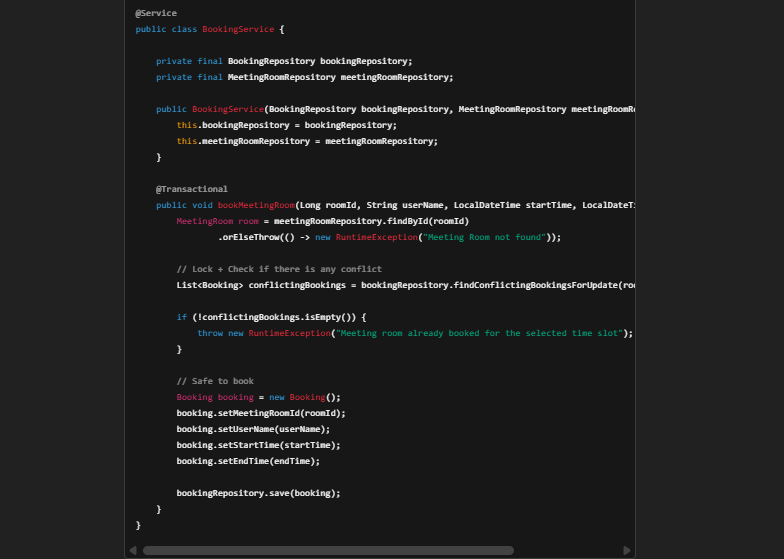
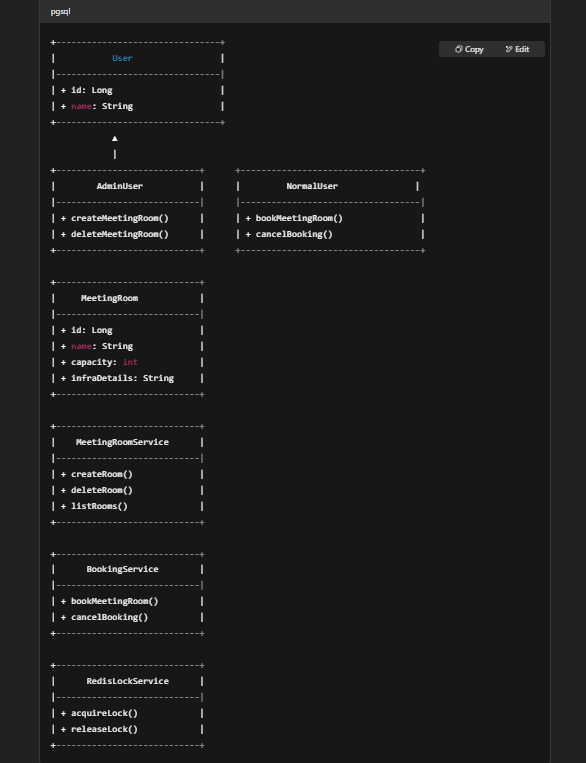
**Meeting room**  
  
How to make it thread Safe when running across multiple servers?   
  
  
Therefor this can be handled 2 ways:  
1. Using a central redis locking  
2. Using SELECT ... FOR UPDATE rightly with Transactional block

**Does @Transactional handle multiple applications/instances?**

**Short Answer:**  
🔹 @Transactional **by itself** does **NOT** guarantee safety across multiple applications or servers.  
🔹 It **only guarantees** ACID **inside the same database** transaction.

**BUT** — if you use **proper database locking** inside that @Transactional block (like SELECT ... FOR UPDATE),  
then **yes**, it **works safely across multiple applications/instances** because the **database itself** is the ultimate lock.  
  
  
Something like:  
  
  
  




✅ **Relationships:**

* AdminUser uses **MeetingRoomService**.
* NormalUser uses **BookingService** (which uses **RedisLockService**).
* **MeetingRoomService** manages meeting rooms.
* **BookingService** manages bookings.