

# Airbnb Internship Spring Boot Assignment

## Assignment Details

### 1. Setup the Project:

- Create a Spring Boot application with Maven/Gradle.
- Add dependencies for Spring Security, JWT, JPA, and a MySQL database.

### 2. Implement Authentication:

- **Signup API:**
  - Endpoint: `POST /api/auth/signup`
  - Request Body: `{ "username": "string", "password": "string", "email": "string", "fullName": "string", "phoneNumber": "string" }`
  - Register new users with encrypted passwords.
  - add some validation(email should uniq, pasword long atleast 8 char)
- **Login API:**
  - Endpoint: `POST /api/auth/login`
  - Request Body: `{ "username": "string", "password": "string" }`
  - Return a JWT token upon successful authentication.

### 3. Property Management:

- **CRUD Operations:**
  - **Create Property:**
    - Endpoint: `POST /api/properties`
    - Request Body: `{ "name": "string", "description": "string", "address": "string", "pricePerNight": "decimal", "numberOfBedrooms": "int", "numberOfBathrooms": "int", "isAvailable": "boolean", "drinkAllowed": "boolean", "petAllowed": "boolean", "maxCheckoutTimeInNights": "int", "extraCharges": "decimal" }`
  - **Retrieve All Properties:**
    - Endpoint: `GET /api/properties`
  - **Get Property by ID:**
    - Endpoint: `GET /api/properties/{id}`
  - **Update Property:**
    - Endpoint: `PUT /api/properties/{id}`
    - Request Body: `{ "name": "string", "description": "string", "address": "string", "pricePerNight": "decimal", "numberOfBedrooms": "int", "numberOfBathrooms": "int", "isAvailable": "boolean", "drinkAllowed": "boolean", "petAllowed": "boolean", "maxCheckoutTimeInNights": "int", "extraCharges": "decimal" }`
  - **Delete Property:**

- Endpoint: DELETE /api/properties/{id}
- **Get User's Properties:**
  - Endpoint: GET /api/users/{userId}/properties
  - Returns all properties owned by the specified user.

## 4. Define the Data Models:

### Property Model:

```
javaCopy code
@Entity
public class Property {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
    private String description;
    private String address;
    private BigDecimal pricePerNight;
    private int numberOfBedrooms;
    private int numberOfBathrooms;
    private boolean isAvailable;
    private boolean drinkAllowed;
    private boolean petAllowed;
    private int maxCheckoutTimeInNights;
    private BigDecimal extraCharges;

    @OneToMany(mappedBy = "property")
    private List<Review> reviews;

    @ManyToOne
    @JoinColumn(name = "owner_id")
    private User owner;
}
```

### User Model:

```
javaCopy code
@Entity
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(unique = true, nullable = false)
    private String username;

    @Column(nullable = false)
    private String password;

    @Column(nullable = false)
    private String email;
}
```

```
@Column(nullable = false)
private String fullName;

private String phoneNumber;

@OneToMany(mappedBy = "owner")
private List<Property> properties;

@OneToMany(mappedBy = "user")
private List<Review> reviews;
}
```

## 5. Implement JWT Authentication:

- Configure Spring Security to use JWT for securing endpoints.
- Create a filter for validating JWT tokens.

---

### Submission Requirements:

- Fully functional Spring Boot application with the implemented features.
- Handle Exception Properly
- Use of appropriate design patterns and best practices.
- A link to the GitHub repository with your project.

**Deadline:** 05-08-2024 by 06pm

**Submission Form :** <https://forms.gle/CTNgwyzb4QypP9Mo9>

### Any Question Regarding Project Or Assignment

whatsapp : 9023379136

email : codewithzosh@gmail.com

---

Good luck, and we look forward to seeing your implementations!