**ShowtimeAPI - Unit Tests Specification**

This document outlines the **unit and integration tests** for the ShowtimeAPI, ensuring both **successful and failing scenarios** are properly covered.

**✅ Success Scenarios**

**1. Add Showtime Successfully**

* **Endpoint:** POST /showtimes
* **Request Body:** Valid ShowtimeRequest with movieId, theater, startTime, endTime, price.
* **Expected:**
  + Status: 200 OK
  + JSON with created Showtime
  + Exists in DB and \_allShowtimes map.

**2. Get Showtime by ID (Cached)**

* **Endpoint:** GET /showtimes/{id}
* **Precondition:** Showtime exists and is in cache.
* **Expected:**
  + Status: 200 OK
  + Correct JSON content

**3. Get Showtime by ID (DB)**

* **Precondition:** Showtime exists only in DB, not in map.
* **Expected:**
  + Status: 200 OK
  + Cache updated

**4. Update Showtime**

* **Endpoint:** POST /showtimes/update/{id}
* **Expected:**
  + Status: 200 OK
  + Fields updated
  + Cache updated

**5. Delete Showtime**

* **Endpoint:** DELETE /showtimes/{id}
* **Expected:**
  + Status: 200 OK
  + Not found in DB or cache after

**❌ Failure Scenarios**

**6. Add Showtime with Missing Fields**

* **Input:** Null/blank fields (e.g., null theater)
* **Expected:** 400 Bad Request + message

**7. Add Showtime for Non-existent Movie**

* **Input:** movieId does not exist in DB
* **Expected:** 404 Not Found

**8. Add Showtime Overlapping**

* **Input:** New showtime overlapping existing one in same theater
* **Expected:** 400 Bad Request + message

**9. Update Showtime with Invalid Data**

* **Input:** End before start, or invalid price
* **Expected:** 400 Bad Request

**10. Update Showtime Not Found**

* **Input:** ID that doesn't exist
* **Expected:** 404 Not Found

**11. Delete Showtime Not Found**

* **Input:** ID not in cache and DB
* **Expected:** 404 Not Found

**12. Get Showtime with Invalid ID Format**

* **Endpoint:** /showtimes/abc
* **Expected:** 400 Bad Request + informative error

**Notes**

* Tests should validate **state in DB** and in \_allShowtimes map.
* Add log verification for coverage.
* Use @Transactional in tests to roll back state between runs.