# Test Suite Document for User Service

## 📌 Overview

| **Item** | **Details** |
| --- | --- |
| System Under Test | UserService (Spring Boot Microservice) |
| Type | REST + Business Logic + Kafka Events + DB Integration |
| Tools | JUnit 5, Mockito, SpringBootTest, Testcontainers |

## 🧪 Test Suite Structure

### 1. Unit Tests

* ✅ Scope: Controllers, Services, Utils
* 🔧 Tools: JUnit 5 + Mockito

### 2. Integration Tests

* ✅ Scope: REST endpoints + DB + Kafka
* 🔧 Tools: SpringBootTest, TestContainers

### 3. Contract Tests *(Optional)*

* ✅ With API Gateway or Consumer

### 4. Security Tests

* ✅ Token validation, permission enforcement

### 5. Negative/Boundary Tests

* ✅ Invalid inputs, missing roles, malformed data

## 🧪 1. Unit Test Cases

### 📂 UserServiceTest.java

| **Test Case** | **Description** |
| --- | --- |
| createUser\_validData | Create a user with valid input |
| createUser\_existingEmail | Email already exists |
| assignRole\_valid | Assign a role to an existing user |
| assignRole\_userNotFound | Fail on missing user |
| getUserProfile\_valid | Fetch profile with valid ID |

## 🧪 2. Controller Test Cases

### 📂 UserControllerTest.java

| **Endpoint** | **Test Case** | **Expected Response** |
| --- | --- | --- |
| POST /user/register | valid input | 201 CREATED |
|  | missing fields / invalid email | 400 BAD REQUEST |
| POST /user/assign-role | user and role valid | 200 OK |
|  | user not found | 404 NOT FOUND |
| GET /user/{id} | existing user | 200 OK |
|  | non-existent ID | 404 NOT FOUND |

## 🧪 3. Security Test Cases

| **Test Case** | **Description** |
| --- | --- |
| access\_with\_valid\_token | Success |
| access\_with\_expired\_token | 401 Unauthorized |
| access\_with\_invalid\_role | 403 Forbidden |
| register\_user\_without\_auth | Should allow (if public endpoint) |

## 🧪 4. Integration Test with Kafka + DB

| **Test Case** | **Description** |
| --- | --- |
| user\_created\_event\_emitted | After registration, check Kafka topic |
| role\_assigned\_event\_emitted | After assigning role |
| DB\_persistence\_check | User saved and fetched from database |
| event\_schema\_validation | Validate payload format sent to Kafka |

## 🛠 Use Testcontainers for Kafka + PostgreSQL in test environment.

## 🧪 5. Negative & Edge Cases

| **Test Case** | **Description** |
| --- | --- |
| createUser\_invalid\_email\_format | Expect 400 BAD REQUEST |
| assignRole\_missing\_user\_id | Expect 400 BAD REQUEST |
| getUser\_invalid\_uuid\_format | Expect 400 BAD REQUEST |
| overly\_large\_input\_values | Expect input validation failure |

## ✅ Tools & Libraries

* **Spring Boot Test**
* **JUnit 5**
* **Mockito / MockMvc**
* **TestContainers (Kafka + Postgres)**
* **RestAssured** (for API-level testing)
* **AssertJ** (for fluent assertions)

## 📦 Folder Structure (example)

bash

src/test/java/

├── controller/

│ └── UserControllerTest.java

├── service/

│ └── UserServiceTest.java

├── integration/

│ └── UserKafkaIntegrationTest.java

├── security/

│ └── AuthTokenFilterTest.java

## 📄 Sample Test Case (JUnit + MockMvc)

java

@Test

void registerUser\_shouldReturnCreated() throws Exception {

UserDto dto = new UserDto("john@example.com", "password123");

mockMvc.perform(post("/user/register")

.contentType(MediaType.APPLICATION\_JSON)

.content(objectMapper.writeValueAsString(dto)))

.andExpect(status().isCreated());

}

## 📊 Summary Table

| **Category** | **#Tests** |
| --- | --- |
| Unit | 10+ |
| Controller/API | 10+ |
| Integration (Kafka, DB) | 6+ |
| Security | 5+ |
| Negative/Edge | 10+ |

# 1. Unit Test Cases

| **Test Case** | **Description** |
| --- | --- |
| createUser\_validData | Create a user with valid input |
| createUser\_existingEmail | Handle duplicate email |
| assignRole\_valid | Assign role to user |
| assignRole\_userNotFound | Fail if user doesn’t exist |
| getUserProfile\_valid | Fetch valid user profile |

## 2. Controller Test Cases

* Validate endpoint responses
* Test status codes
* Ensure correct validation behavior

## 3. Security Test Cases

* ✅ access\_with\_valid\_token – Success
* 🔐 access\_with\_expired\_token – 401 Unauthorized
* ❌ access\_with\_invalid\_role – 403 Forbidden
* 🌐 register\_user\_without\_auth – Allowed if endpoint is public

## 4. Integration Tests (Kafka + DB)

* 📤 user\_created\_event\_emitted – On user creation
* 🧾 role\_assigned\_event\_emitted – On role assignment
* 💾 DB\_persistence\_check – Check DB for saved user
* 🔍 event\_schema\_validation – Validate Kafka payload format

## 5. Negative & Edge Case Tests

* ❌ createUser\_invalid\_email\_format – 400 BAD REQUEST
* ⛔ assignRole\_missing\_user\_id – 400 BAD REQUEST
* ⚠ getUser\_invalid\_uuid\_format – 400 BAD REQUEST
* 💥 overly\_large\_input\_values – Should be rejected