

Alumni Network Test Cases

Overview

The testing suite covers both the **controller layer** and the **service layer** of a Spring Boot application. These tests use **JUnit 5** along with **Mockito** for mocking dependencies. The tests ensure that core functionalities—from user management to work experience, achievements, events, emails, and real-time chat—operate correctly under both normal and edge conditions.

Controller Tests

1. UserControllerTest

- **Registration & Login:** Tests successful user registration (with password encryption and point assignment) and login (with both valid and invalid credentials).
- **User Retrieval:** Ensures that getting all users and fetching a user by ID work as expected.
- **Session & Logout:** Verifies session-based operations like logout.

2. SearchControllerTest

- **Search Functionality:** Validates searching by various criteria (name, company, batch) and tests scenarios with empty results or missing parameters.

3. ContactMessageControllerTest

- **Form Submission:** Confirms that submitting a contact message saves the message.
- **Status Updates & Deletion:** Tests marking messages as resolved and deleting messages, including proper handling of not-found cases.

4. EmailControllerTest

- **Password Reset & Invitations:** Verifies sending reset emails, handling missing users, and sending invitations with multiple recipients (including edge cases like empty email addresses).

5. GalleryControllerTest

- **Folder & Image Management:** Tests retrieval of gallery folders, image upload (with both valid and missing fields), and deletion of folders and images.

6. GroupControllerTest

- **Chat Group Operations:** Checks group creation, joining, exiting, and retrieving recent groups. Uses mock authentication with `@WithMockUser`.
7. **JobOpportunityControllerTest**
 - **Job Management:** Validates retrieval of job opportunities, addition of new job posts (with point adjustments), and deletion before/after deadlines, including not-found scenarios.
 8. **UserPointsControllerTest**
 - **Leaderboard & Points:** Tests endpoints to fetch user points and display the leaderboard, including scenarios where points may be zero or the leaderboard is empty.
 9. **WebSocketControllerTest**
 - **Real-Time Chat Monitoring:** Verifies that the WebSocket controller correctly logs connected users and handles exceptions during user retrieval.
 10. **WorkExperienceControllerTest**
 - **Work Experience CRUD:** Ensures adding, retrieving (by user and by ID), updating, and deleting work experience records work correctly.
-

Service Tests

1. **UserServiceTest**
 - **User Operations:** Tests registration (success and duplicate email), login (with valid/invalid credentials), user retrieval (all users and by ID), updating profiles, and deleting users.
2. **WorkExperienceServiceTest**
 - **Work Experience Management:** Covers adding a work experience, retrieving by user, getting all experiences, updating, and deletion. Also includes handling scenarios where a user is not found or a record is missing.
3. **AchievementServiceTest**
 - **Achievements CRUD:** Tests retrieving all achievements, adding a new achievement (with point assignment), fetching by user or by ID (both found and not found), updating achievements, and deletion.
4. **EmailServiceTest**

- **Email Operations:** Validates sending password reset emails using SimpleMailMessage and sending invites via MIME messages. Includes edge cases for missing or empty email addresses.

5. **EventServiceTest**

- **Event Management:** Tests retrieving all events, creating an event, and deleting an event.

6. **AlumniApplicationTests**

- **Context Load:** Ensures the Spring Boot application context loads successfully without errors.

Tools & Frameworks

- **JUnit 5** for writing tests.
- **Mockito** for creating mocks of dependencies.
- **Spring Boot Test** for initializing web contexts in controller tests.
- **MockMvc** for simulating HTTP requests to controllers.
- **JavaMailSender** for email service testing.
- **BCryptPasswordEncoder** to simulate password encryption.