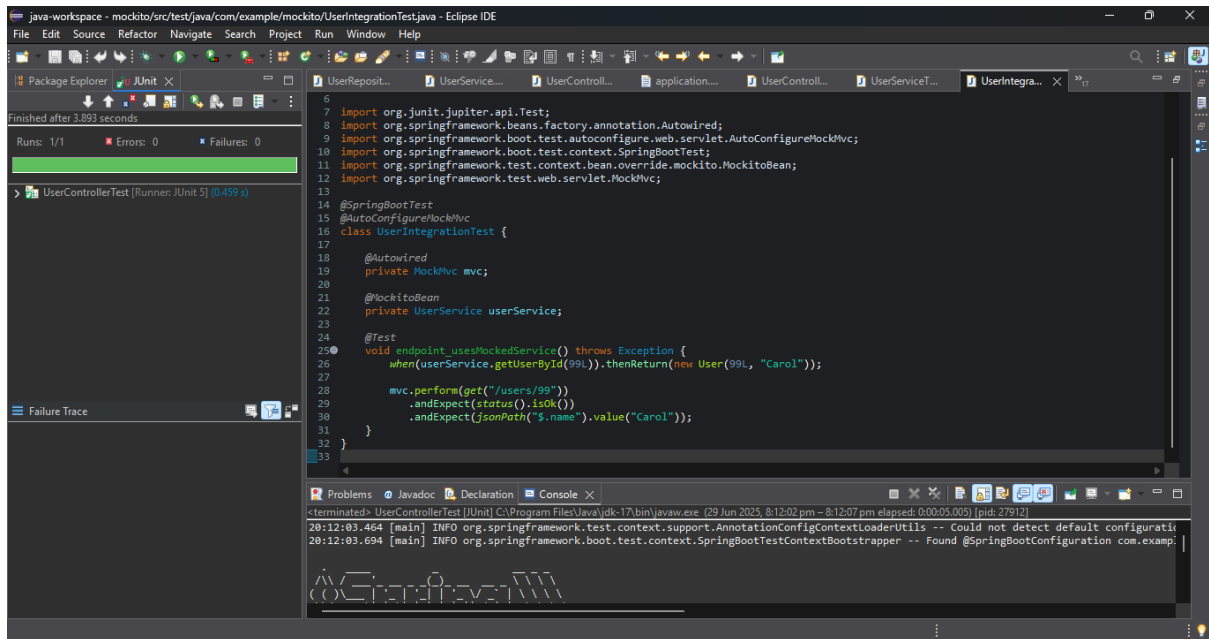


Mocking Dependencies in Spring Tests using Mockito Output

Exercise 1: Mocking a Service Dependency in a Controller Test

Output:



```
6 import org.junit.jupiter.api.Test;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
9 import org.springframework.boot.test.context.SpringBootTest;
10 import org.springframework.test.context.bean.override.mockito.MockitoBean;
11 import org.springframework.test.web.servlet.MockMvc;
12
13
14 @SpringBootTest
15 @AutoConfigureMockMvc
16 class UserIntegrationTest {
17
18     @Autowired
19     private MockMvc mvc;
20
21     @MockBean
22     private UserService userService;
23
24     @Test
25     void endpoint_usesMockedService() throws Exception {
26         when(userService.getUserById(99L)).thenReturn(new User(99L, "Carol"));
27
28         mvc.perform(get("/users/99"))
29             .andExpect(status().isOk())
30             .andExpect(jsonPath("$.name").value("Carol"));
31     }
32 }
33
```

Finished after 3.893 seconds

Runs: 1/1 Errors: 0 Failures: 0

UserIntegrationTest [Runner: JUnit 5] (0.459 s)

Problems | Javadoc | Declaration | Console

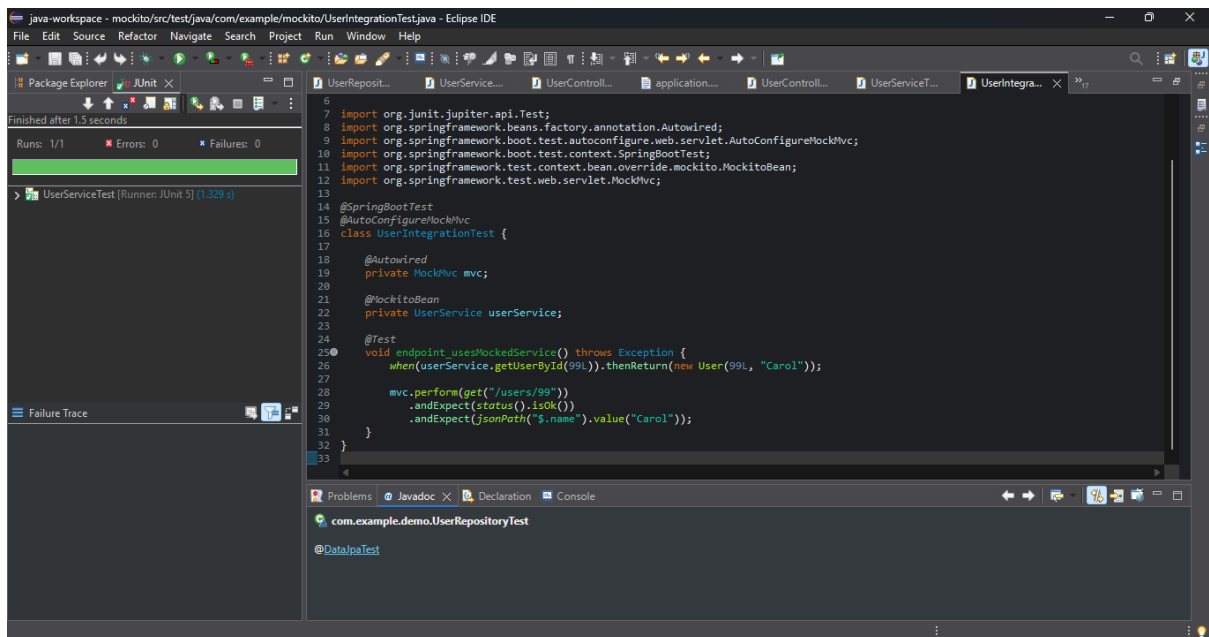
20:12:03.464 [main] INFO org.springframework.boot.test.context.support.AnnotationConfigContextLoaderUtils -- Could not detect default configurati

20:12:03.694 [main] INFO org.springframework.boot.test.context.SpringBootTest -- Found @SpringBootTestConfiguration com.examp

Mocking the service layer via `@WebMvcTest` + `@MockBean` lets you focus purely on request/response behaviour without spinning up business logic.

Exercise 2: Mocking a Repository in a Service Test

Output:



```
6 import org.junit.jupiter.api.Test;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;
9 import org.springframework.boot.test.context.SpringBootTest;
10 import org.springframework.test.context.bean.override.mockito.MockitoBean;
11 import org.springframework.test.web.servlet.MockMvc;
12
13
14 @SpringBootTest
15 @AutoConfigureMockMvc
16 class UserIntegrationTest {
17
18     @Autowired
19     private MockMvc mvc;
20
21     @MockBean
22     private UserService userService;
23
24     @Test
25     void endpoint_usesMockedService() throws Exception {
26         when(userService.getUserById(99L)).thenReturn(new User(99L, "Carol"));
27
28         mvc.perform(get("/users/99"))
29             .andExpect(status().isOk())
30             .andExpect(jsonPath("$.name").value("Carol"));
31     }
32 }
33
```

Finished after 1.5 seconds

Runs: 1/1 Errors: 0 Failures: 0

UserIntegrationTest [Runner: JUnit 5] (1.329 s)

Problems | Javadoc | Declaration | Console

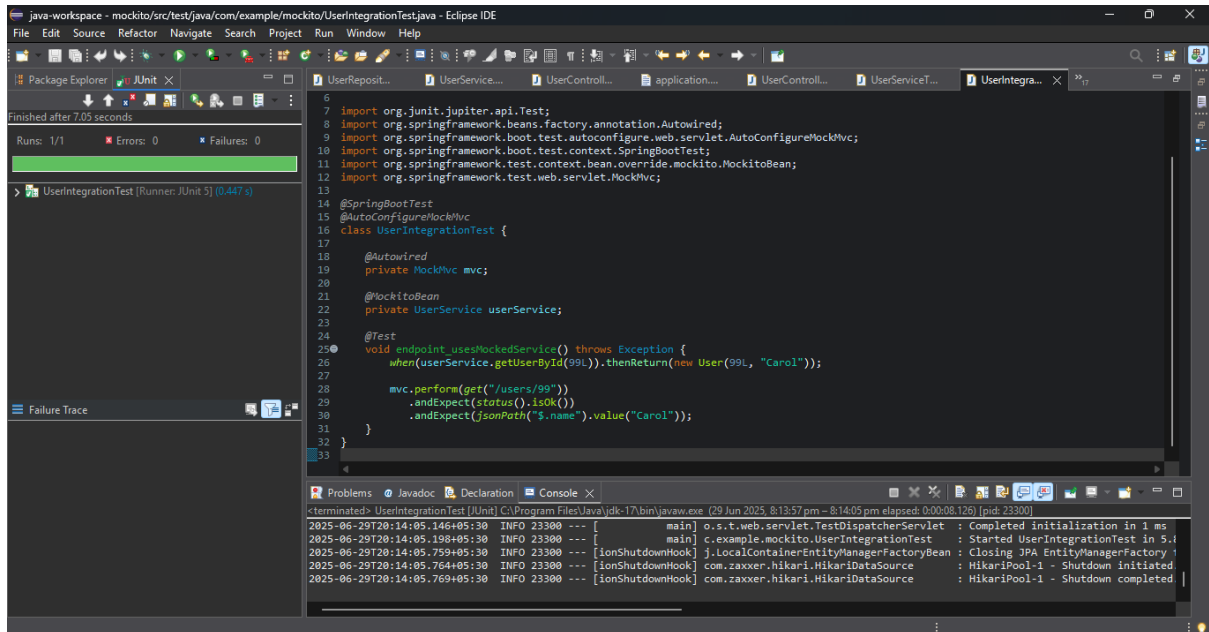
com.example.demo.UserRepositoryTest

@DataJpaTest

Replacing the repository with a Mockito mock (`@ExtendWith(MockitoExtension.class)`) isolates business rules from persistence and keeps tests lightning-fast.

Exercise 3: Mocking a Service Dependency in an Integration Test

Output:



```
java-workspace - mockito/src/test/java/com/example/mockito/UserIntegrationTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit X
Finished after 7.05 seconds
Runs: 1/1 Errors: 0 Failures: 0
> UserIntegrationTest [Runner: JUnit 5] (0.447 s)

class UserIntegrationTest {
    @Autowired
    private MockMvc mvc;

    @MockitoBean
    private UserService userService;

    @Test
    void endpoint_userMockedService() throws Exception {
        when(userService.getUserById(99L)).thenReturn(new User(99L, "Carol"));

        mvc.perform(get("/users/99"))
            .andExpect(status().isOk())
            .andExpect(jsonPath("$.name").value("Carol"));
    }
}

Problems Javadoc Declaration Console
terminated: UserIntegrationTest [JUnit] C:\Program Files\Java\jdk-17\bin\javaw.exe (29 Jun 2025 8:13:57 pm - 8:14:05 pm elapsed: 0:00:08.126 [pid: 23300])
2025-06-29T20:14:05.146+05:30 INFO 23300 --- [main] o.s.v.t.web.servlet.TestDispatcherServlet : Completed initialization in 1 ms
2025-06-29T20:14:05.198+05:30 INFO 23300 --- [main] c.example.mockito.UserIntegrationTest : Started UserIntegrationTest in 5.1
2025-06-29T20:14:05.759+05:30 INFO 23300 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory !
2025-06-29T20:14:05.764+05:30 INFO 23300 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated.
2025-06-29T20:14:05.769+05:30 INFO 23300 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
```

In a full-context test (`@SpringBootTest`), mock only the external or slow components, e.g. remote services, to verify wiring while maintaining speed and determinism.