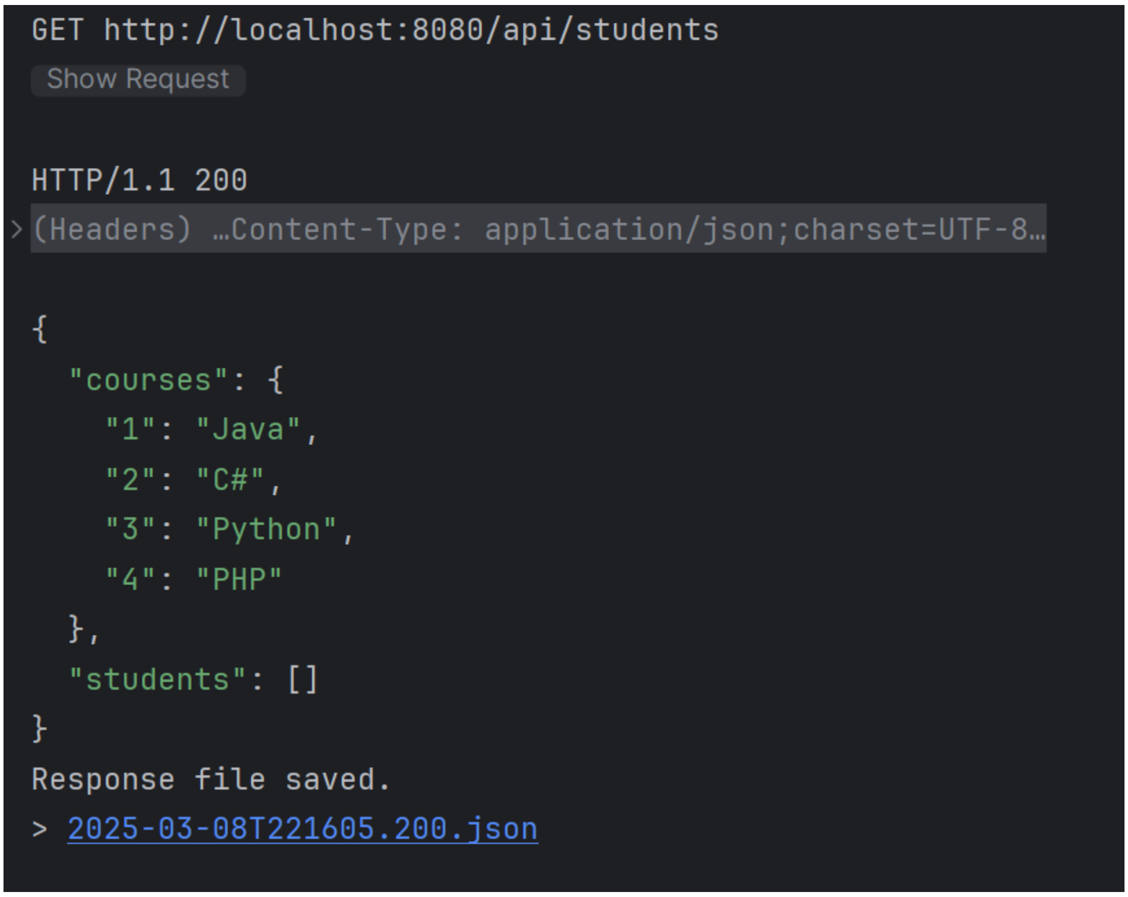
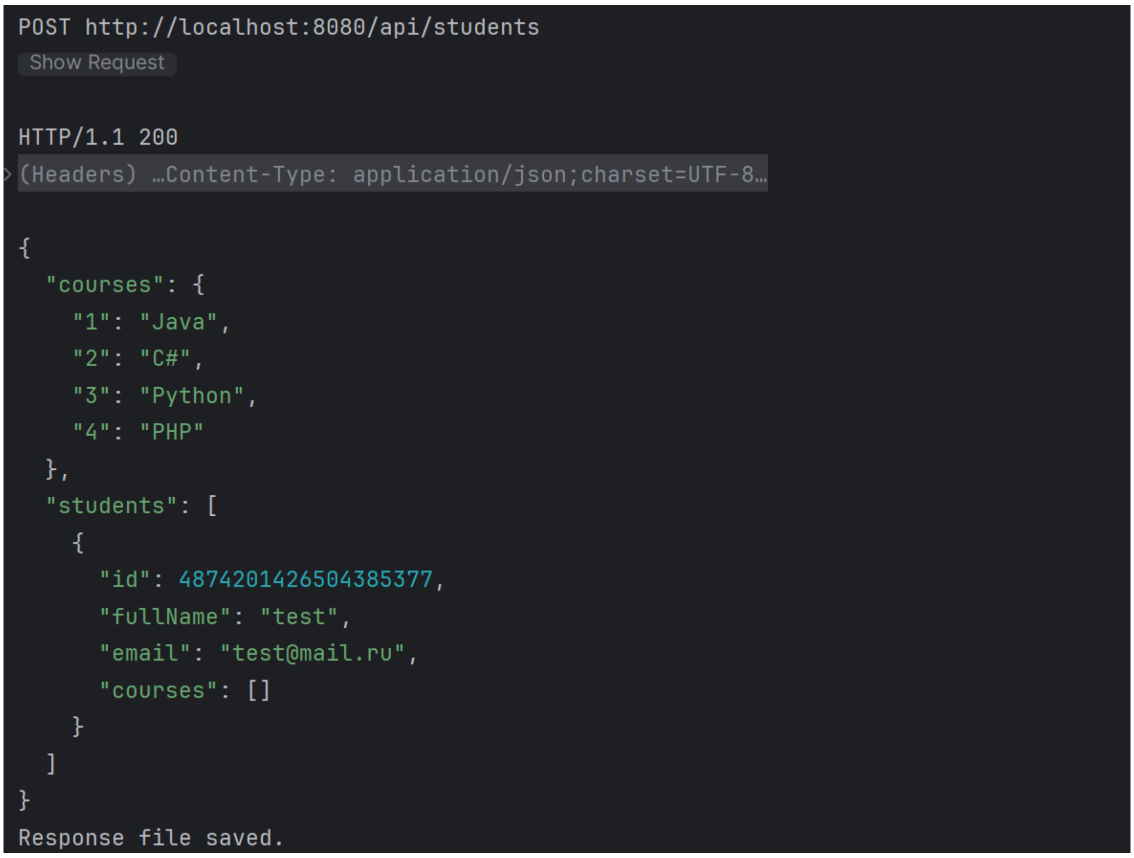
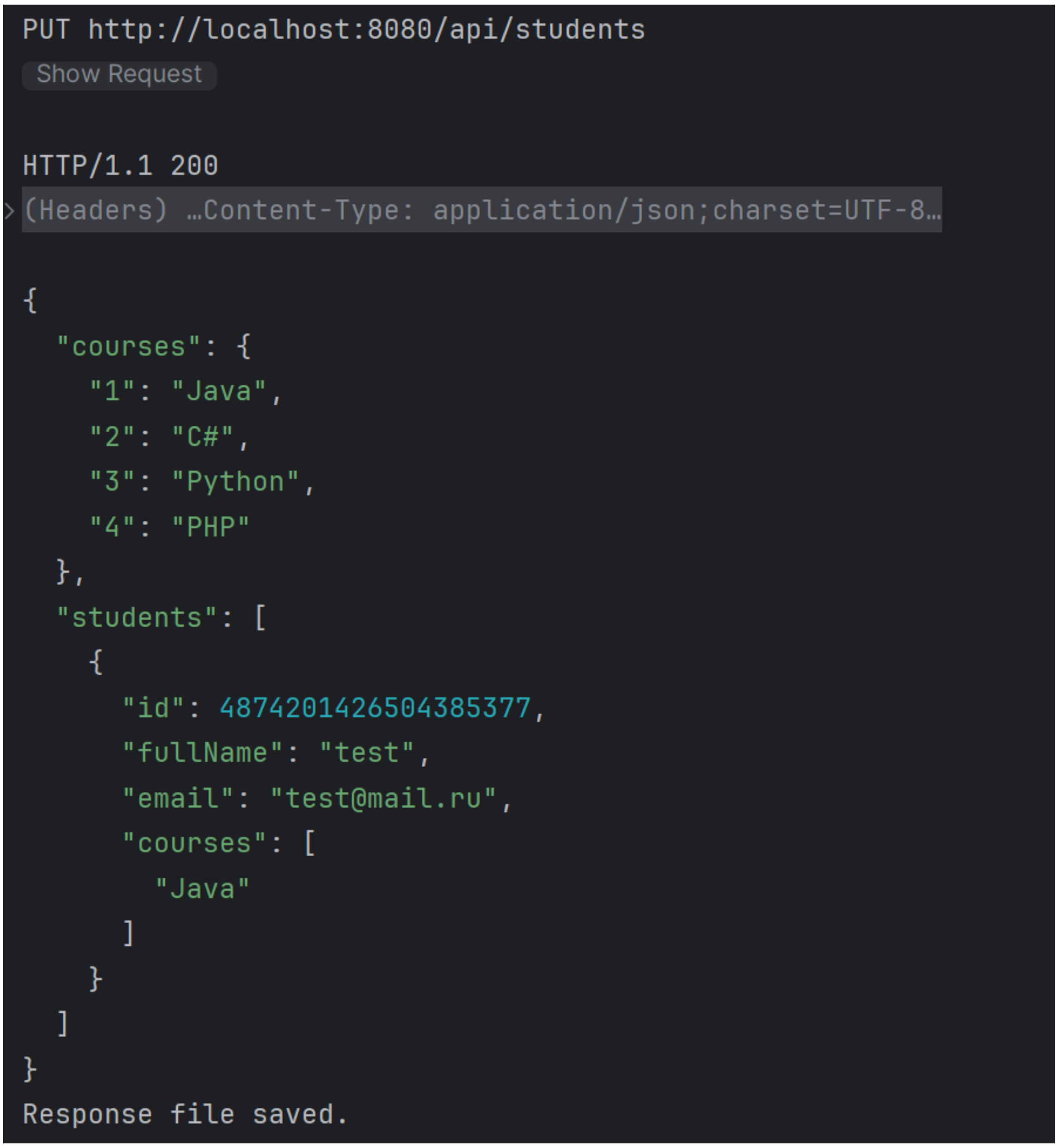
Задание1. Студент может зарегистрироваться на курс



```
GET http://localhost:8080/api/students
Show Request

HTTP/1.1 200
> (Headers) …Content-Type: application/json;charset=UTF-8…

{
  "courses": {
    "1": "Java",
    "2": "C#",
    "3": "Python",
    "4": "PHP"
  },
  "students": []
}
Response file saved.
> 2025-03-08T221605.200.json
```

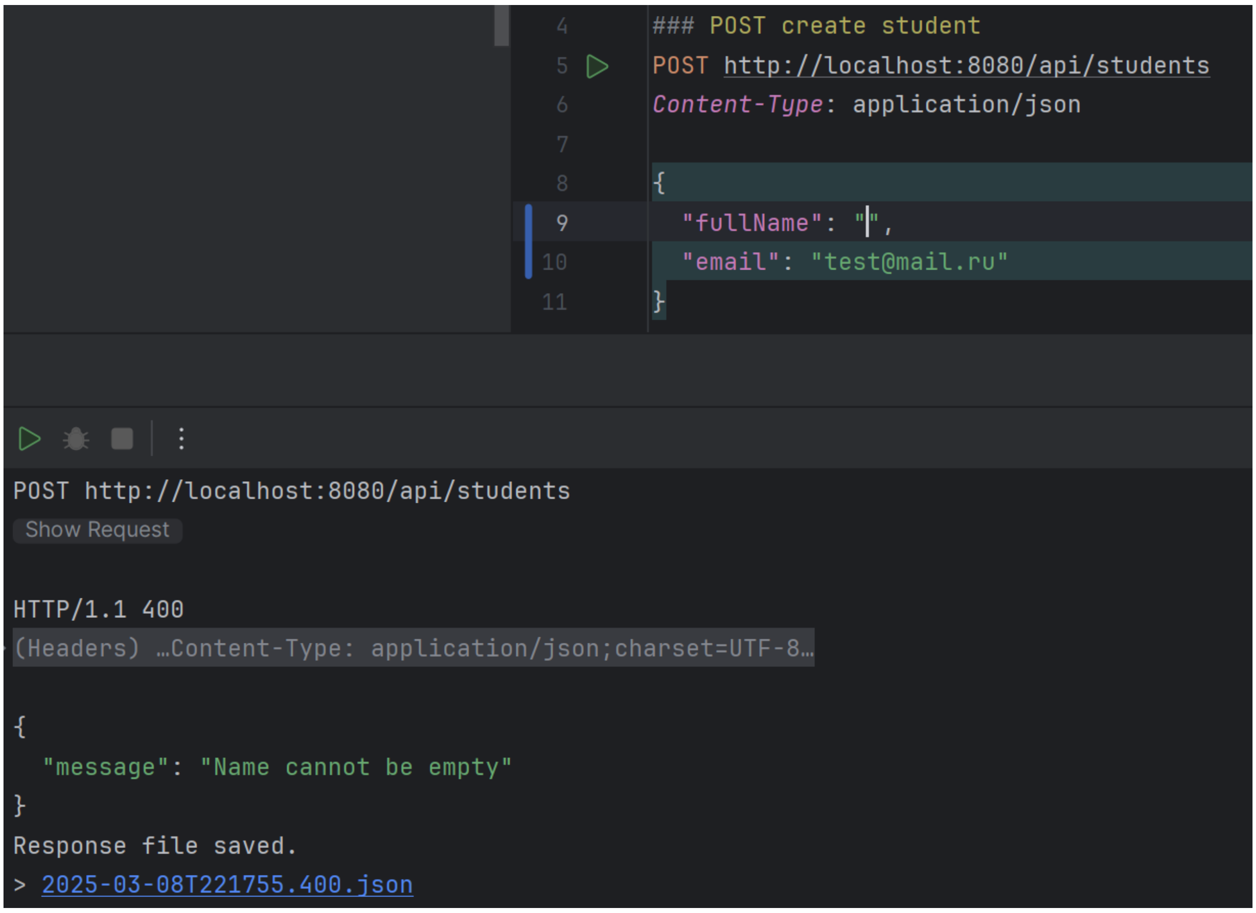


```
POST http://localhost:8080/api/students
Show Request

HTTP/1.1 200
> (Headers) …Content-Type: application/json;charset=UTF-8…

{
  "courses": {
    "1": "Java",
    "2": "C#",
    "3": "Python",
    "4": "PHP"
  },
  "students": [
    {
      "id": 4874201426504385377,
      "fullName": "test",
      "email": "test@mail.ru",
      "courses": []
    }
  ]
}
Response file saved.
```



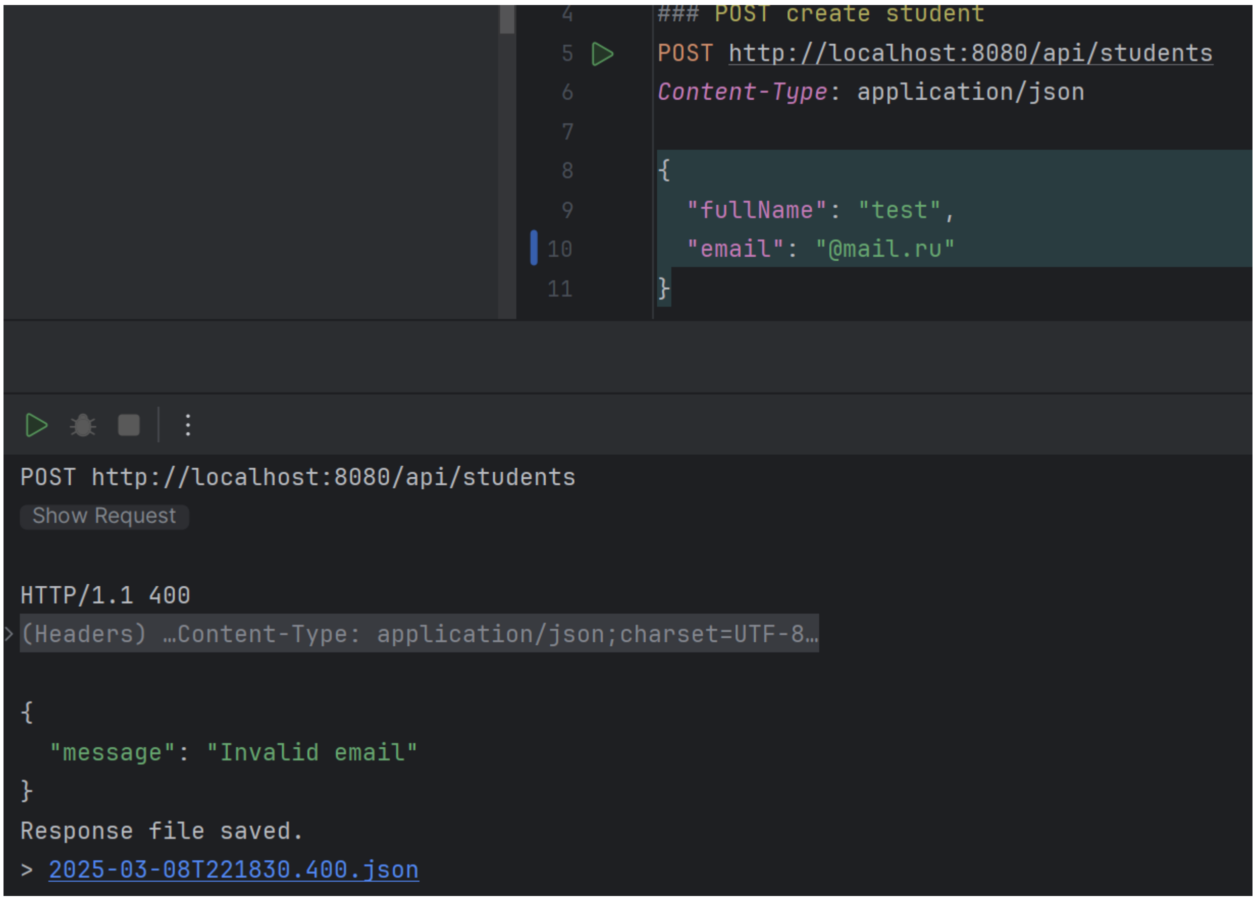
```
PUT http://localhost:8080/api/students
Show Request

HTTP/1.1 200
(Headers) …Content-Type: application/json;charset=UTF-8…

{
  "courses": {
    "1": "Java",
    "2": "C#",
    "3": "Python",
    "4": "PHP"
  },
  "students": [
    {
      "id": 4874201426504385377,
      "fullName": "test",
      "email": "test@mail.ru",
      "courses": [
        "Java"
      ]
    }
  ]
}
Response file saved.
```

Задание2. Все вводимые параметры должны иметь проверки валидности



```
### POST create student
POST http://localhost:8080/api/students
Content-Type: application/json

{
  "fullName": "",
  "email": "test@mail.ru"
}
```

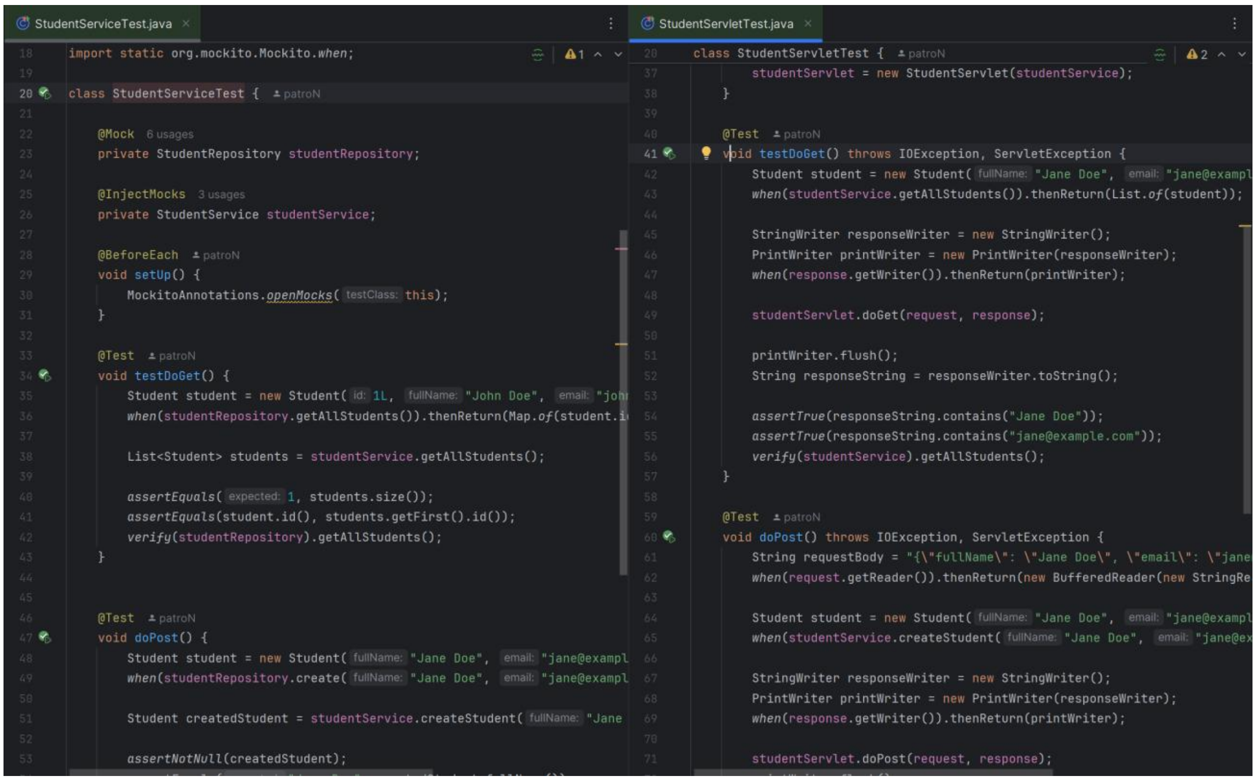
```
POST http://localhost:8080/api/students
Show Request

HTTP/1.1 400
(Headers) …Content-Type: application/json;charset=UTF-8…

{
  "message": "Name cannot be empty"
}
Response file saved.
> 2025-03-08T221755.400.json
```



```
### POST create student
POST http://localhost:8080/api/students
Content-Type: application/json

{
  "fullName": "test",
  "email": "@mail.ru"
}
```

```
POST http://localhost:8080/api/students
Show Request

HTTP/1.1 400
(Headers) …Content-Type: application/json;charset=UTF-8…

{
  "message": "Invalid email"
}
Response file saved.
> 2025-03-08T221830.400.json
```

Задание 3. Тесты должны проверять работу всех операций

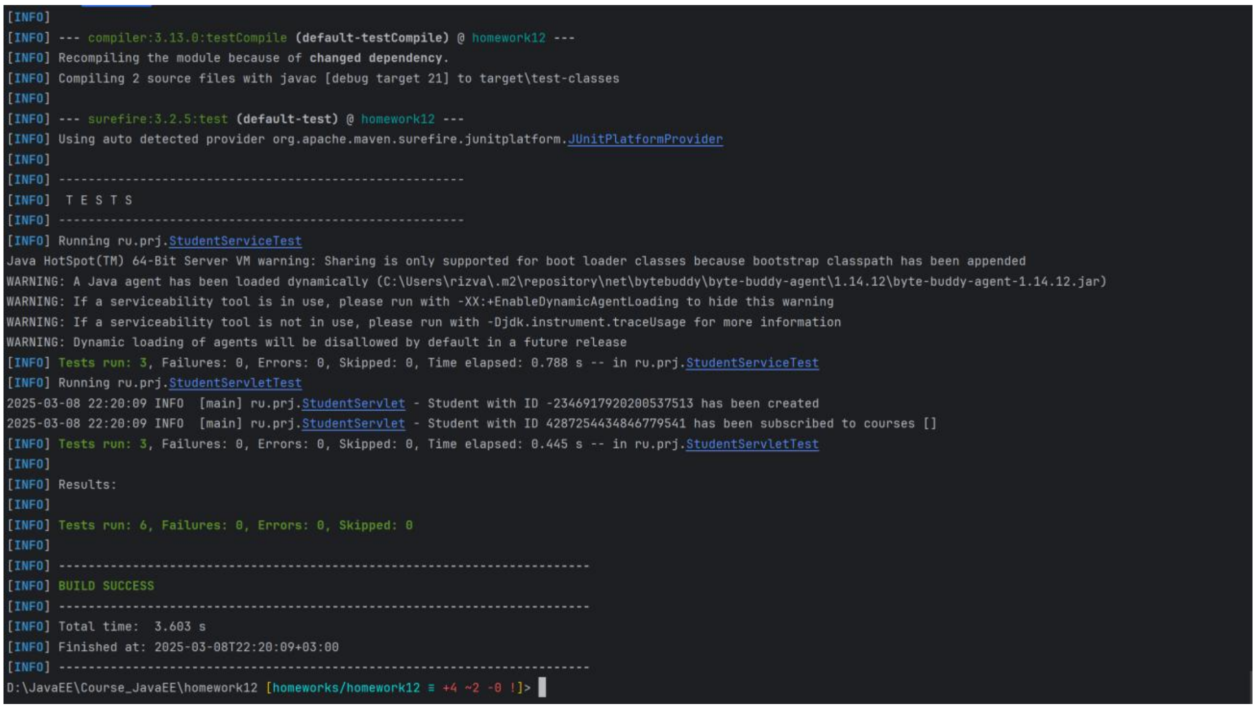


StudentServiceTest.java

```java
import static org.mockito.Mockito.when;

class StudentServiceTest {

    @Mock
    private StudentRepository studentRepository;

    @InjectMocks
    private StudentService studentService;

    @BeforeEach
    void setUp() {
        MockitoAnnotations.openMocks(this);
    }

    @Test
    void testDoGet() {
        Student student = new Student(1L, "John Doe", "joh
        when(studentRepository.getAllStudents()).thenReturn(Map.of(student.i

        List<Student> students = studentService.getAllStudents();

        assertEquals(1, students.size());
        assertEquals(student.id(), students.getFirst().id());
        verify(studentRepository).getAllStudents();
    }


    @Test
    void doPost() {
        Student student = new Student("Jane Doe", "jane@exampl
        when(studentRepository.create("Jane Doe", "jane@exampl

        Student createdStudent = studentService.createStudent("Jane

        assertNotNull(createdStudent);
```

StudentServletTest.java

```java
class StudentServletTest {
        studentServlet = new StudentServlet(studentService);
    }

    @Test
    void testDoGet() throws IOException, ServletException {
        Student student = new Student("Jane Doe", "jane@exampl
        when(studentService.getAllStudents()).thenReturn(List.of(student));

        StringWriter responseWriter = new StringWriter();
        PrintWriter printWriter = new PrintWriter(responseWriter);
        when(response.getWriter()).thenReturn(printWriter);

        studentServlet.doGet(request, response);

        printWriter.flush();
        String responseString = responseWriter.toString();

        assertTrue(responseString.contains("Jane Doe"));
        assertTrue(responseString.contains("jane@example.com"));
        verify(studentService).getAllStudents();
    }

    @Test
    void doPost() throws IOException, ServletException {
        String requestBody = "{\"fullName\": \"Jane Doe\", \"email\": \"jane
        when(request.getReader()).thenReturn(new BufferedReader(new StringRe

        Student student = new Student("Jane Doe", "jane@exampl
        when(studentService.createStudent("Jane Doe", "jane@ex

        StringWriter responseWriter = new StringWriter();
        PrintWriter printWriter = new PrintWriter(responseWriter);
        when(response.getWriter()).thenReturn(printWriter);

        studentServlet.doPost(request, response);
```



```
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ homework12 ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 2 source files with javac [debug target 21] to target\test-classes
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ homework12 ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running ru.prj.StudentServiceTest
Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
WARNING: A Java agent has been loaded dynamically (C:\Users\rizva\.m2\repository\net\bytebuddy\byte-buddy-agent\1.14.12\byte-buddy-agent-1.14.12.jar)
WARNING: If a serviceability tool is in use, please run with -XX:+EnableDynamicAgentLoading to hide this warning
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.788 s -- in ru.prj.StudentServiceTest
[INFO] Running ru.prj.StudentServletTest
2025-03-08 22:20:09 INFO  [main] ru.prj.StudentServlet - Student with ID -2346917920200537513 has been created
2025-03-08 22:20:09 INFO  [main] ru.prj.StudentServlet - Student with ID 4287254434846779541 has been subscribed to courses []
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.445 s -- in ru.prj.StudentServletTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.603 s
[INFO] Finished at: 2025-03-08T22:20:09+03:00
[INFO] ------------------------------------------------------------------------
D:\JavaEE\Course_JavaEE\homework12 [homeworks/homework12 ≡ +4 ~2 -0 !]>
```