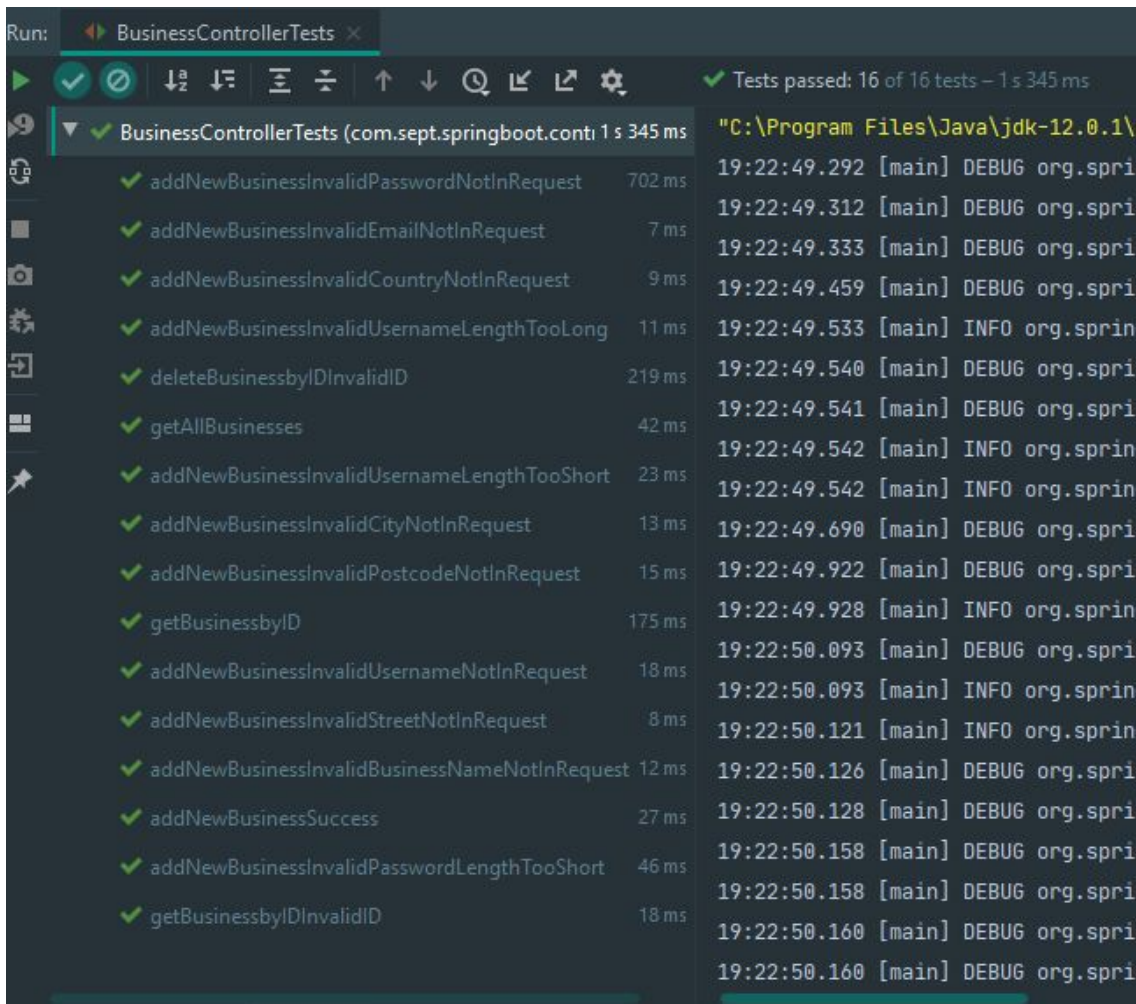
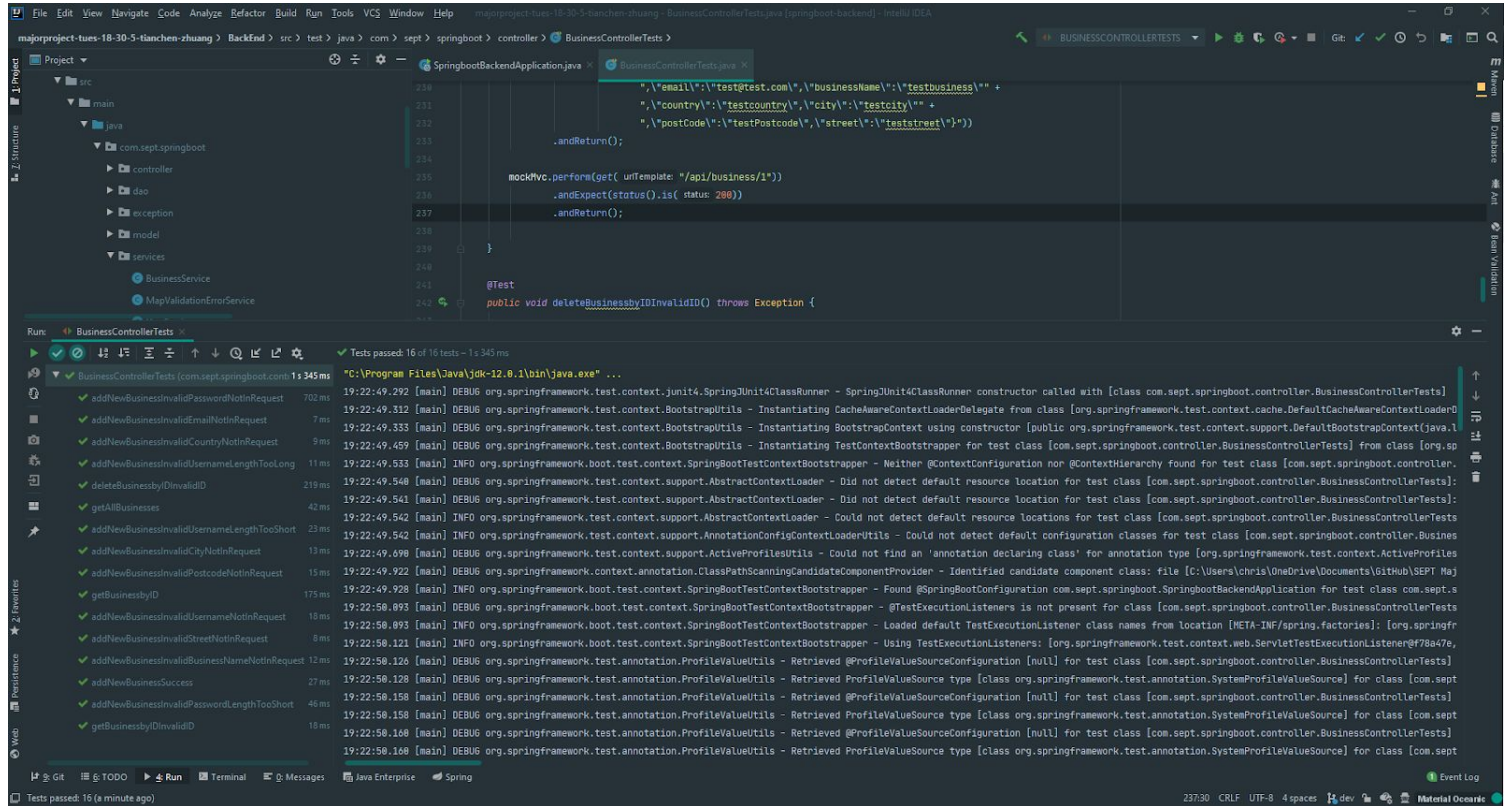


Unit Test Execution Screenshots

Business Controller Tests:



User Controller Tests:

The screenshot shows the IntelliJ IDEA IDE with the `UserControllerTests.java` file open. The code defines a `@Test` method `addNewUserInvalidEmailNotInRequest` that uses `MockMvc` to perform a POST request to `/api/user` with a JSON body containing a password and a username. The test expects a 400 status and a specific JSON response indicating an email error.

```
185         .andExpect(content().json( jsonContent: "{\"password\": \"Password is required\"}"))
186         .andReturn();
187     }
188
189     @Test
190     public void addNewUserInvalidEmailNotInRequest() throws Exception {
191
192         mockMvc.perform(
193             post( uriTemplate: "/api/user")
194                 .contentType(MediaType.APPLICATION_JSON_VALUE)
195                 .content("{\"password\":\"testpassword\",\"username\":\"testusername\"}"))
196             .andExpect(status().is( 400))
197             .andExpect(content().json( jsonContent: "{\"email\": \"Email is required\"}"))
198             .andReturn();
199     }
200 }
```

The Run console at the bottom shows the test results for `UserControllerTests`. All 7 tests passed in 579 ms.

Test Name	Duration	Time	Log Level	Message
UserControllerTests (com.sept.springboot.controller)	579 ms	2020-09-21 19:27:35.527	INFO 10536	main] o.s.b.t.w.SpringBootMockServletContext : Initializing Spring TestDispatcherServlet ''
addNewUserInvalidUsernameNotInRequest	450 ms	2020-09-21 19:27:35.527	INFO 10536	main] o.s.t.web.servlet.TestDispatcherServlet : Initializing Servlet ''
addNewUserInvalidUsernameLengthTooLong	10 ms	2020-09-21 19:27:35.529	INFO 10536	main] o.s.t.web.servlet.TestDispatcherServlet : Completed initialization in 2 ms
addNewUserInvalidPasswordNotInRequest	9 ms			
addNewUserInvalidUsernameLengthTooShort	11 ms			
addNewUserSuccess	84 ms			
addNewUserInvalidEmailNotInRequest	6 ms			
addNewUserInvalidPasswordLengthTooShort	9 ms			

This is a close-up view of the Run console from the previous screenshot, showing the detailed test results for `UserControllerTests`. The tests are listed with their durations and the console output for the first three tests.

Test Name	Duration	Time	Log Level	Message
UserControllerTests (com.sept.springboot.controller)	579 ms	2020-09-21 19:27:35.527	INFO 10536	main] o.s.b.t.w.SpringBootMockServletContext : Initializing Spring TestDispatcherServlet ''
addNewUserInvalidUsernameNotInRequest	450 ms	2020-09-21 19:27:35.527	INFO 10536	main] o.s.t.web.servlet.TestDispatcherServlet : Initializing Servlet ''
addNewUserInvalidUsernameLengthTooLong	10 ms	2020-09-21 19:27:35.529	INFO 10536	main] o.s.t.web.servlet.TestDispatcherServlet : Completed initialization in 2 ms
addNewUserInvalidPasswordNotInRequest	9 ms			
addNewUserInvalidUsernameLengthTooShort	11 ms			
addNewUserSuccess	84 ms			
addNewUserInvalidEmailNotInRequest	6 ms			
addNewUserInvalidPasswordLengthTooShort	9 ms			

FrontEnd Tests:

```
//test if business exists
describe("check if business list isn't empty",()=>{
  it("test if business is exists",()=>{
    let bList = [{ 'title': 'res', 'id': 1 }];
    let item = mount(<BusinessList></BusinessList>)
    item.setState({list:bList})
    expect(item.find("ClassCard")!=null)
  })
})
//mounting 2 business lists
describe("test mounting business list",()=>{
  it("test mounting two business",()=>{
    let bList = [{ 'title': 'res', 'id': 1 },
    { 'title': 'gym', 'id': 2 }];
    let item = mount(<BusinessList></BusinessList>)
    item.setState({list:bList})
    expect(item.find("ClassCard")).toHaveLength(2)
  })
})
//testing if business list label correctly
describe("testing if business list label exists",()=>{
  it("test business label display correctly",()=>{
    render(<BusinessList />)
    screen.queryAllByLabelText('Businesses list');
  })
})
//testing if img path is correct or exists
describe("check img path is correct or exists",()=>{
  it("check img exists",()=>{
```

```
console.log src/components/booking/businessList.jsx:43
[
  { businessname: 'mrkitchen', bid: 1, desc: 'its kitchen' },
  { businessname: 'gym', bid: 2, desc: 'its gym' }
]
```

```
console.log src/components/booking/businessList.jsx:43
[
  { businessname: 'mrkitchen', bid: 1, desc: 'its kitchen' },
  { businessname: 'gym', bid: 2, desc: 'its gym' }
]
```

```
console.log src/components/booking/businessList.jsx:43
[
  { businessname: 'mrkitchen', bid: 1, desc: 'its kitchen' },
  { businessname: 'gym', bid: 2, desc: 'its gym' }
]
```

Test Suites: 1 passed, 1 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 32.635s
Ran all test suites.

Watch Usage: Press w to show more.[]

```

2 import React from 'react';
3 import {mount,configure} from 'enzyme';
4 import Register from './register';
5 import RegisterB from './registerB';
6 configure({ adapter: new Adapter() });
7
8 //test if the register succeed
9 describe("check if register data has been store in",()=>{
10   it("test if register data actually exists",()=>{
11     let user = [{ 'password': '12345678', 'email': '123456@gmail.com','username': '123456','cpassword':'12345678' }];
12     let console = mount(<Register></Register>)
13     console.setState({input:user})
14     expect(console.find("Register") != null);
15   })
16 })
17
18 //test if the register for business succeed
19 describe("check if register data of a business has been store in",()=>{
20   it("test if register data actually exists",()=>{
21     let user = [{ 'password': '12345678', 'email': '123456business@gmail.com','username': '123456','address':'163 collin street','cpassword':'12345678' }];
22     let console = mount(<RegisterB></RegisterB>)
23     console.setState({input:user})
24     expect(console.find("Register") != null);
25   })
26 })

```

```

1 import Adapter from 'enzyme-adapter-react-16';
2 import React from 'react';
3 import {mount,configure} from 'enzyme';
4 import Login from './login';
5 import LoginB from './loginB';
6 configure({ adapter: new Adapter() });
7
8 //test if the user can succeed to login
9 describe("check if user can succeed to login",()=>{
10   it("test if the data user enter can help them login",()=>{
11     let user = [{ 'password': '12345678', 'email': '123456@gmail.com'}];
12     let console = mount(<Login></Login>)
13     console.setState({input:user})
14     expect(console.find("Login") != null);
15   })
16 })
17
18 //test if the business user can succeed to login
19 describe("check if the business user can succeed to login",()=>{
20   it("test if the data that business user enter can help them login",()=>{
21     let user = [{ 'password': '12345678', 'email': '123456business@gmail.com'}];
22     let console = mount(<LoginB></LoginB>)
23     console.setState({input:user})
24     expect(console.find("LoginB") != null);
25   })
26 })

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PASS src/components/register/register.test.js (5.509s)
PASS src/components/login/login.test.js (5.754s)

Test Suites: 2 passed, 2 total
Tests: 4 passed, 4 total
Snapshots: 0 total
Time: 8.577s
Ran all test suites.

Watch Usage: Press w to show more. ☐