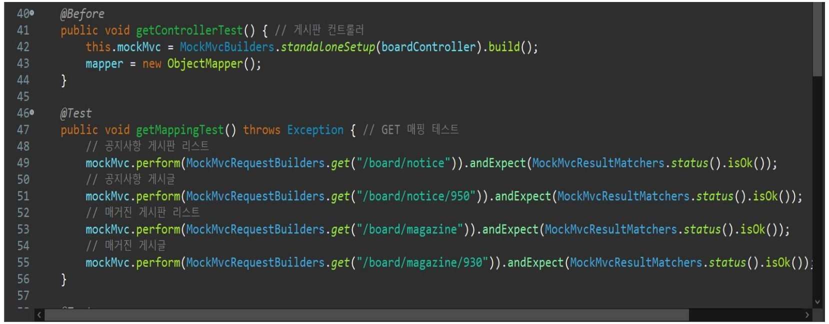
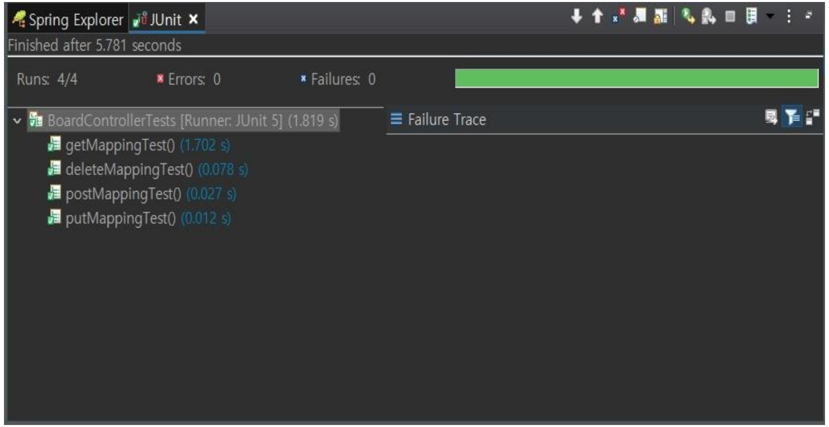
Tetleaf

통합 테스트

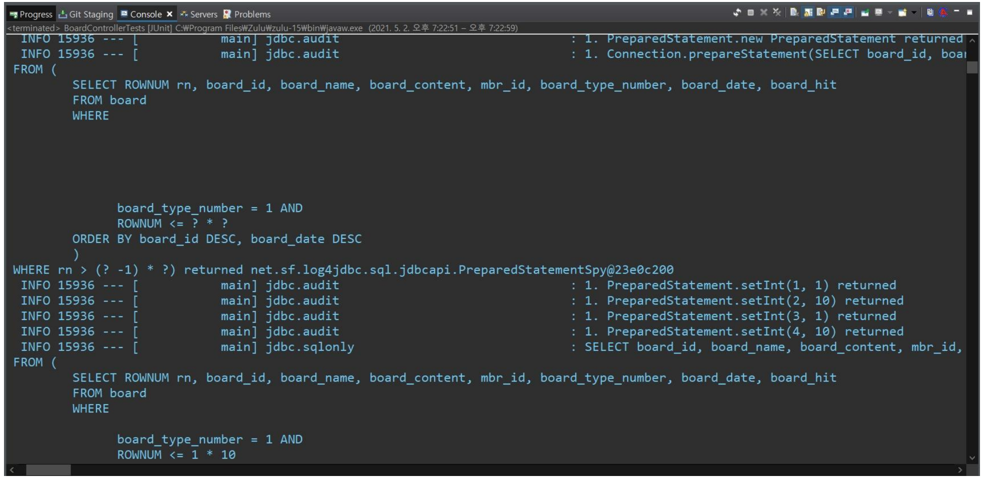
01 Board 통합 테스트



```java
    @Before
    public void getControllerTest() { // 게시판 컨트롤러
        this.mockMvc = MockMvcBuilders.standaloneSetup(boardController).build();
        mapper = new ObjectMapper();
    }

    @Test
    public void getMappingTest() throws Exception { // GET 매핑 테스트
        // 공지사항 게시판 리스트
        mockMvc.perform(MockMvcRequestBuilders.get("/board/notice")).andExpect(MockMvcResultMatchers.status().isOk());
        // 공지사항 게시글
        mockMvc.perform(MockMvcRequestBuilders.get("/board/notice/950")).andExpect(MockMvcResultMatchers.status().isOk());
        // 매거진 게시판 리스트
        mockMvc.perform(MockMvcRequestBuilders.get("/board/magazine")).andExpect(MockMvcResultMatchers.status().isOk());
        // 매거진 게시글
        mockMvc.perform(MockMvcRequestBuilders.get("/board/magazine/930")).andExpect(MockMvcResultMatchers.status().isOk());
    }
```

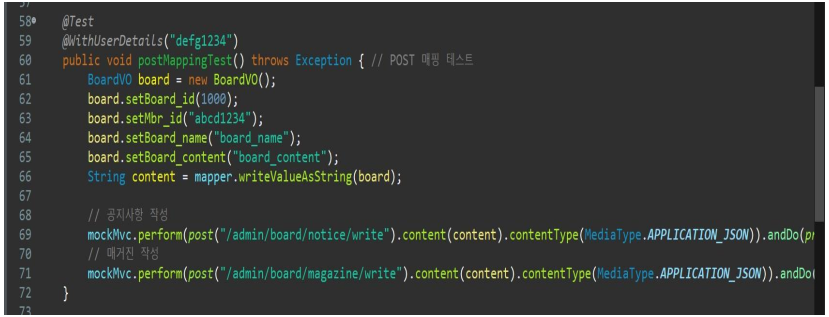


```
Finished after 5.781 seconds

Runs: 4/4        Errors: 0        Failures: 0

BoardControllerTests [Runner: JUnit 5] (1.819 s)
    getMappingTest() (1.702 s)
    deleteMappingTest() (0.078 s)
    postMappingTest() (0.027 s)
    putMappingTest() (0.012 s)
```

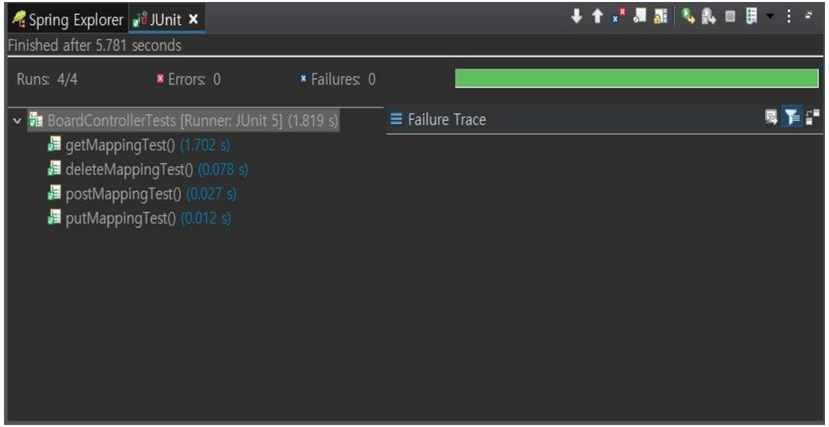


```
<terminated> BoardControllerTests [JUnit] C:₩Program Files₩Zulu₩zulu-15₩bin₩javaw.exe  (2021. 5. 2. 오후 7:22:51 – 오후 7:22:59)
 INFO 15936 --- [           main] jdbc.audit                               : 1. PreparedStatement.new PreparedStatement returned
 INFO 15936 --- [           main] jdbc.audit                               : 1. Connection.prepareStatement(SELECT board_id, boar
FROM (
        SELECT ROWNUM rn, board_id, board_name, board_content, mbr_id, board_type_number, board_date, board_hit
        FROM board
        WHERE




              board_type_number = 1 AND
              ROWNUM <= ? * ?
        ORDER BY board_id DESC, board_date DESC
        )
WHERE rn > (? -1) * ?) returned net.sf.log4jdbc.sql.jdbcapi.PreparedStatementSpy@23e0c200
 INFO 15936 --- [           main] jdbc.audit                               : 1. PreparedStatement.setInt(1, 1) returned
 INFO 15936 --- [           main] jdbc.audit                               : 1. PreparedStatement.setInt(2, 10) returned
 INFO 15936 --- [           main] jdbc.audit                               : 1. PreparedStatement.setInt(3, 1) returned
 INFO 15936 --- [           main] jdbc.audit                               : 1. PreparedStatement.setInt(4, 10) returned
 INFO 15936 --- [           main] jdbc.sqlonly                             : SELECT board_id, board_name, board_content, mbr_id,
FROM (
        SELECT ROWNUM rn, board_id, board_name, board_content, mbr_id, board_type_number, board_date, board_hit
        FROM board
        WHERE

              board_type_number = 1 AND
              ROWNUM <= 1 * 10
```

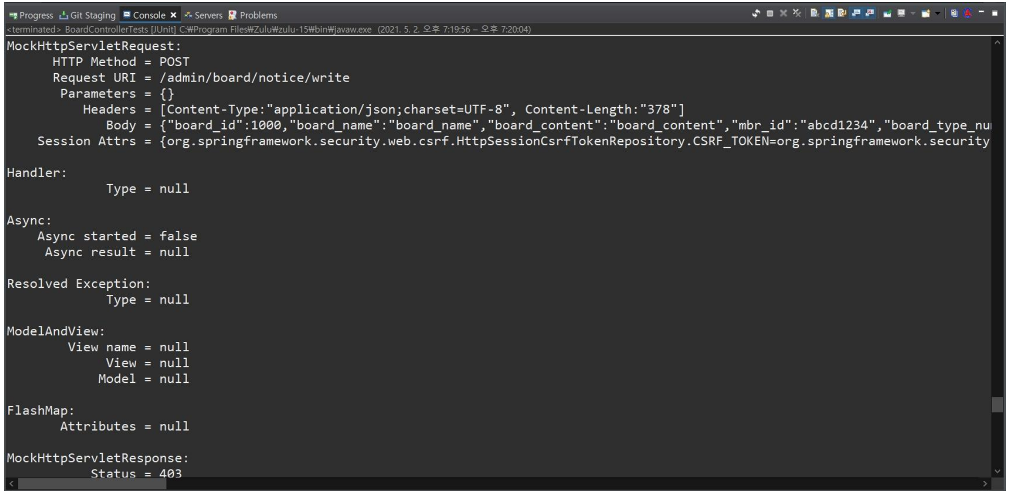
01 Board 통합 테스트



```
    @Test
    @WithUserDetails("defg1234")
    public void postMappingTest() throws Exception { // POST 매핑 테스트
        BoardVO board = new BoardVO();
        board.setBoard_id(1000);
        board.setMbr_id("abcd1234");
        board.setBoard_name("board_name");
        board.setBoard_content("board_content");
        String content = mapper.writeValueAsString(board);

        // 공지사항 작성
        mockMvc.perform(post("/admin/board/notice/write").content(content).contentType(MediaType.APPLICATION_JSON)).andDo(pr
        // 매거진 작성
        mockMvc.perform(post("/admin/board/magazine/write").content(content).contentType(MediaType.APPLICATION_JSON)).andDo(
    }
```

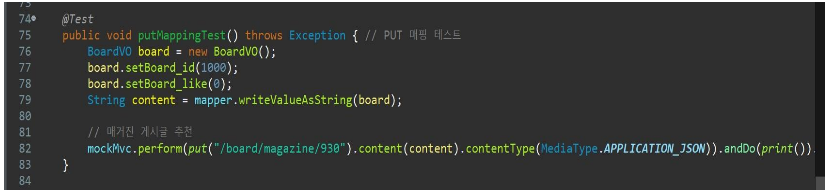


```
Finished after 5.781 seconds
Runs: 4/4    Errors: 0    Failures: 0

BoardControllerTests [Runner: JUnit 5] (1.819 s)
    getMappingTest() (1.702 s)
    deleteMappingTest() (0.078 s)
    postMappingTest() (0.027 s)
    putMappingTest() (0.012 s)
```

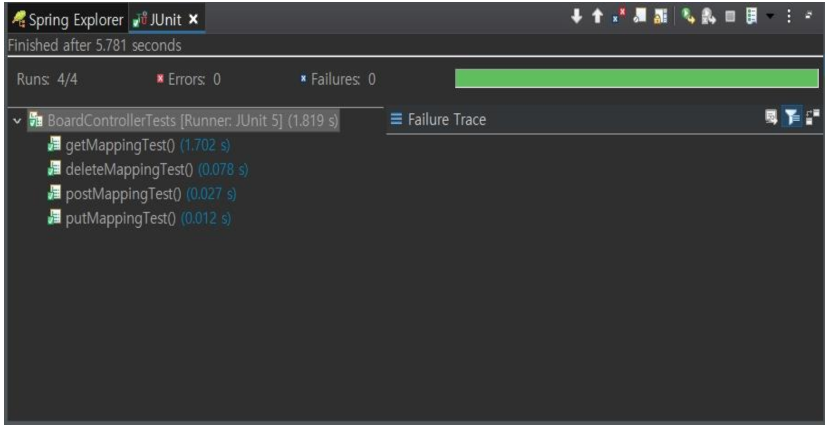


```
<terminated> BoardControllerTests [JUnit] C:₩Program Files₩Zulu₩zulu-15₩bin₩javaw.exe  (2021. 5. 2. 오후 7:19:56 – 오후 7:20:04)
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /admin/board/notice/write
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"378"]
             Body = {"board_id":1000,"board_name":"board_name","board_content":"board_content","mbr_id":"abcd1234","board_type_nu
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
```

01 Board 통합 테스트



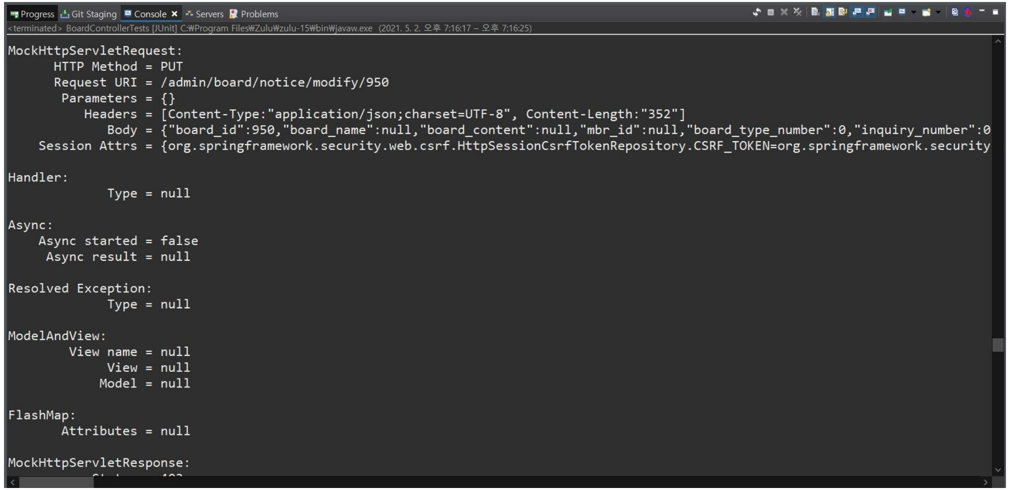
```java
    @Test
    public void putMappingTest() throws Exception { // PUT 매핑 테스트
        BoardVO board = new BoardVO();
        board.setBoard_id(1000);
        board.setBoard_like(0);
        String content = mapper.writeValueAsString(board);

        // 매거진 게시글 추천
        mockMvc.perform(put("/board/magazine/930").content(content).contentType(MediaType.APPLICATION_JSON)).andDo(print()).
    }
```



Finished after 5.781 seconds

Runs: 4/4   Errors: 0   Failures: 0

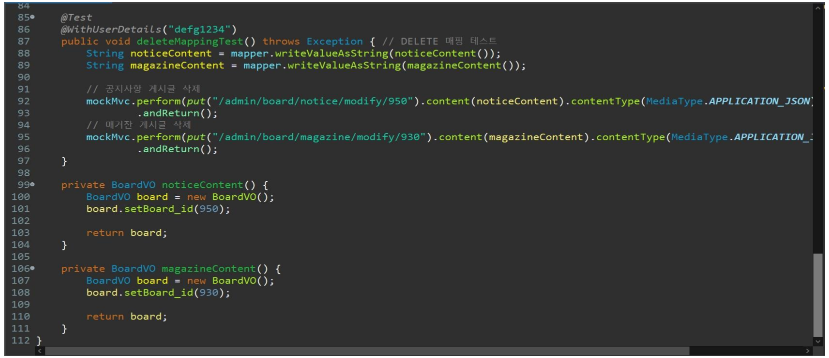
- BoardControllerTests [Runner: JUnit 5] (1.819 s)
  - getMappingTest() (1.702 s)
  - deleteMappingTest() (0.078 s)
  - postMappingTest() (0.027 s)
  - putMappingTest() (0.012 s)



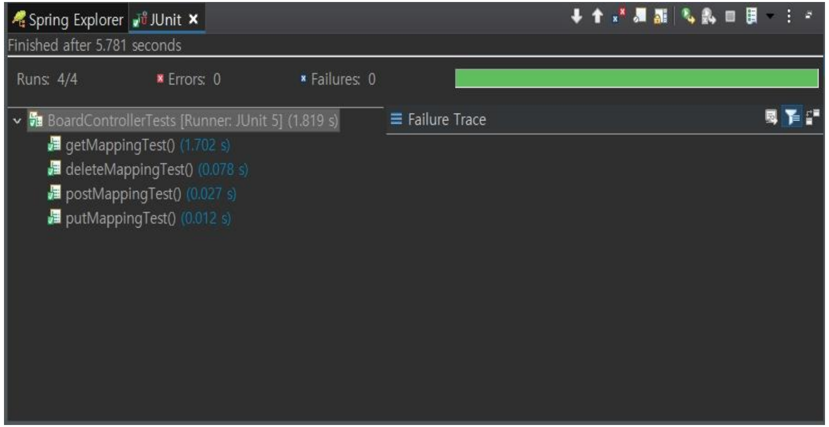
<terminated> BoardControllerTests [JUnit] C:₩Program Files₩Zulu₩zulu-15₩bin₩javaw.exe (2021. 5. 2. 오후 7:16:17 – 오후 7:16:25)

```
MockHttpServletRequest:
      HTTP Method = PUT
      Request URI = /admin/board/notice/modify/950
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"352"]
             Body = {"board_id":950,"board_name":null,"board_content":null,"mbr_id":null,"board_type_number":0,"inquiry_number":0
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
```

01 Board 통합 테스트



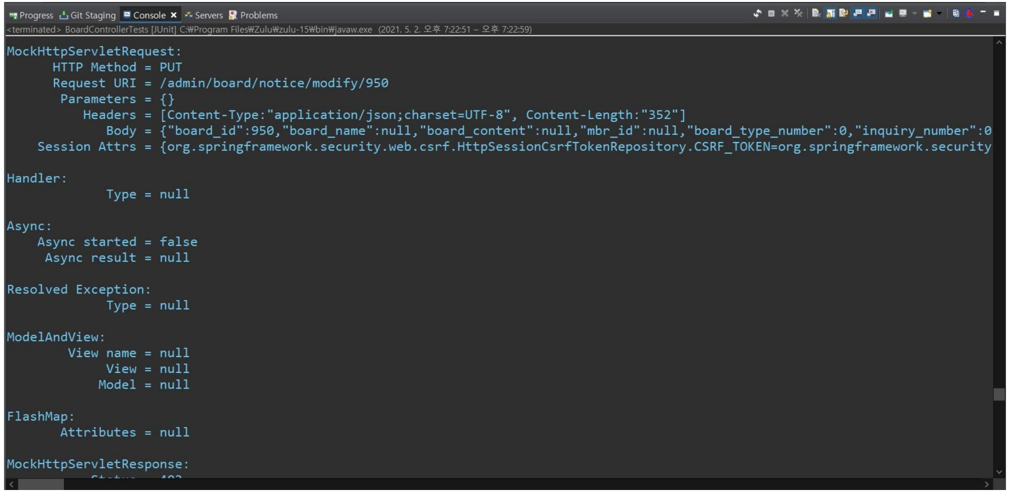
```java
    @Test
    @WithUserDetails("defg1234")
    public void deleteMappingTest() throws Exception { // DELETE 매핑 테스트
        String noticeContent = mapper.writeValueAsString(noticeContent());
        String magazineContent = mapper.writeValueAsString(magazineContent());

        // 공지사항 게시글 삭제
        mockMvc.perform(put("/admin/board/notice/modify/950").content(noticeContent).contentType(MediaType.APPLICATION_JSON)
                .andReturn();
        // 매거진 게시글 삭제
        mockMvc.perform(put("/admin/board/magazine/modify/930").content(magazineContent).contentType(MediaType.APPLICATION_J
                .andReturn();
    }

    private BoardVO noticeContent() {
        BoardVO board = new BoardVO();
        board.setBoard_id(950);

        return board;
    }

    private BoardVO magazineContent() {
        BoardVO board = new BoardVO();
        board.setBoard_id(930);

        return board;
    }
}
```



Finished after 5.781 seconds

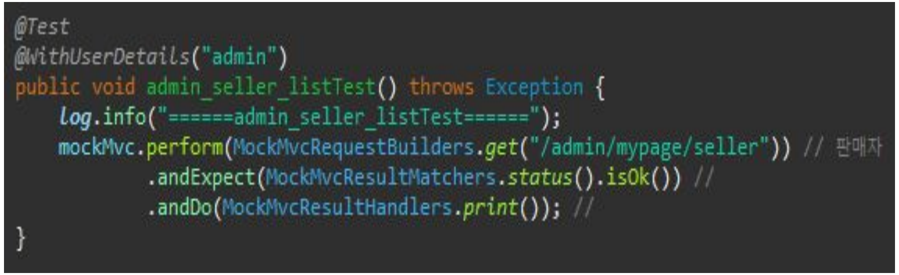
Runs: 4/4　Errors: 0　Failures: 0

- BoardControllerTests [Runner: JUnit 5] (1.819 s)
  - getMappingTest() (1.702 s)
  - deleteMappingTest() (0.078 s)
  - postMappingTest() (0.027 s)
  - putMappingTest() (0.012 s)

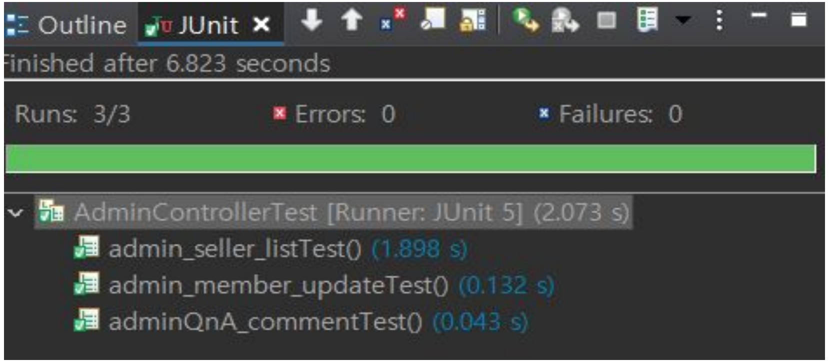


```
<terminated> BoardControllerTests [JUnit] C:₩Program Files₩Zulu₩zulu-15₩bin₩javaw.exe  (2021. 5. 2. 오후 7:22:51 – 오후 7:22:59)
MockHttpServletRequest:
      HTTP Method = PUT
      Request URI = /admin/board/notice/modify/950
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"352"]
             Body = {"board_id":950,"board_name":null,"board_content":null,"mbr_id":null,"board_type_number":0,"inquiry_number":0
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
```

02 Admin 통합 테스트



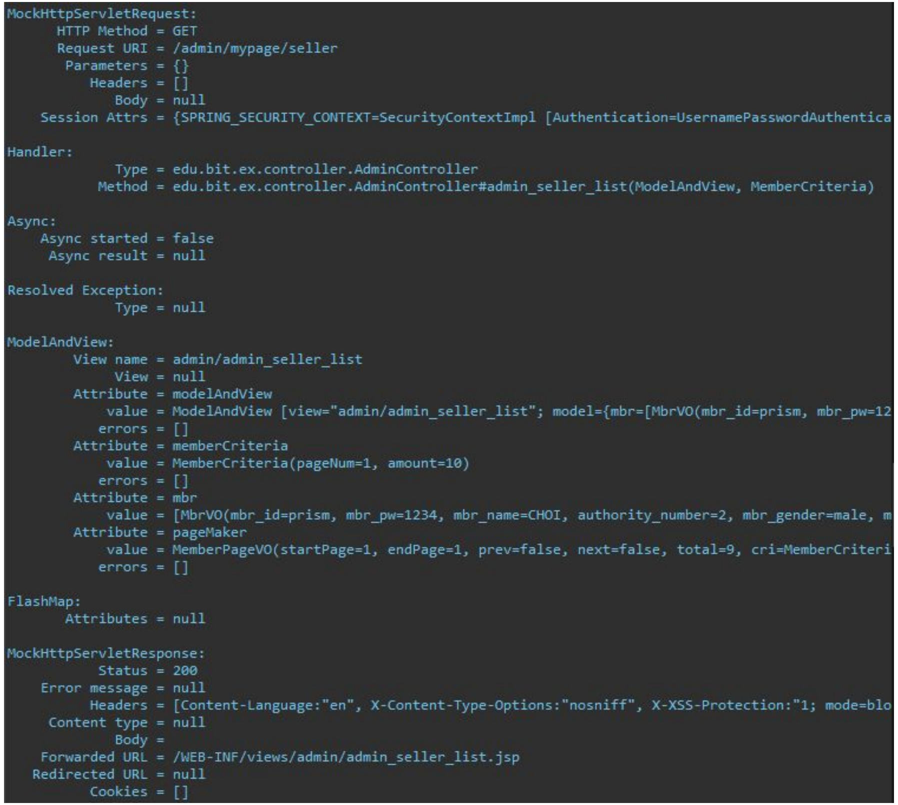
```
@Test
@WithUserDetails("admin")
public void admin_seller_listTest() throws Exception {
    log.info("======admin_seller_listTest======");
    mockMvc.perform(MockMvcRequestBuilders.get("/admin/mypage/seller")) // 판매자
            .andExpect(MockMvcResultMatchers.status().isOk()) //
            .andDo(MockMvcResultHandlers.print()); //
}
```



Finished after 6.823 seconds

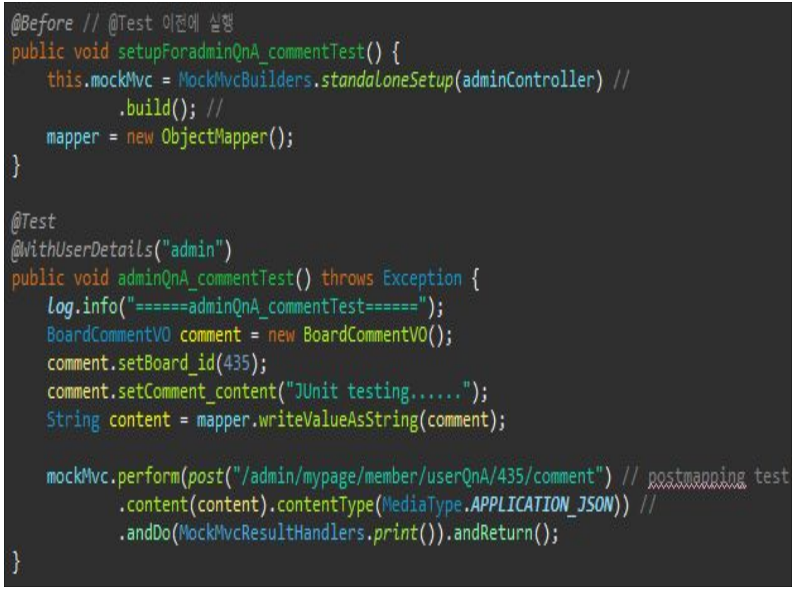
Runs: 3/3 Errors: 0 Failures: 0

- AdminControllerTest [Runner: JUnit 5] (2.073 s)
  - admin_seller_listTest() (1.898 s)
  - admin_member_updateTest() (0.132 s)
  - adminQnA_commentTest() (0.043 s)

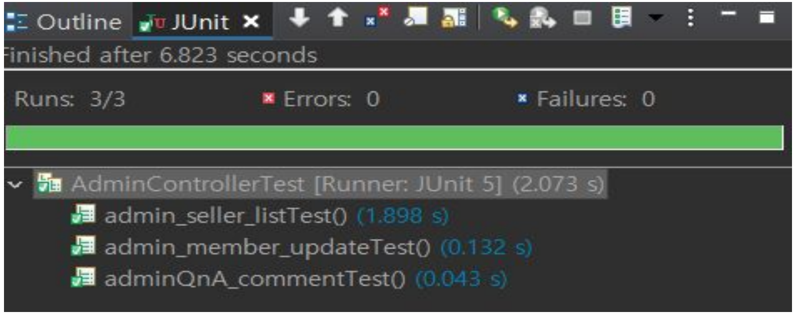


```
MockHttpServletRequest:
      HTTP Method = GET
      Request URI = /admin/mypage/seller
       Parameters = {}
          Headers = []
             Body = null
    Session Attrs = {SPRING_SECURITY_CONTEXT=SecurityContextImpl [Authentication=UsernamePasswordAuthentica

Handler:
             Type = edu.bit.ex.controller.AdminController
           Method = edu.bit.ex.controller.AdminController#admin_seller_list(ModelAndView, MemberCriteria)

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = admin/admin_seller_list
             View = null
        Attribute = modelAndView
            value = ModelAndView [view="admin/admin_seller_list"; model={mbr=[MbrVO(mbr_id=prism, mbr_pw=12
           errors = []
        Attribute = memberCriteria
            value = MemberCriteria(pageNum=1, amount=10)
           errors = []
        Attribute = mbr
            value = [MbrVO(mbr_id=prism, mbr_pw=1234, mbr_name=CHOI, authority_number=2, mbr_gender=male, m
        Attribute = pageMaker
            value = MemberPageVO(startPage=1, endPage=1, prev=false, next=false, total=9, cri=MemberCriteri
           errors = []

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 200
    Error message = null
          Headers = [Content-Language:"en", X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=blo
     Content type = null
             Body = 
    Forwarded URL = /WEB-INF/views/admin/admin_seller_list.jsp
   Redirected URL = null
          Cookies = []
```

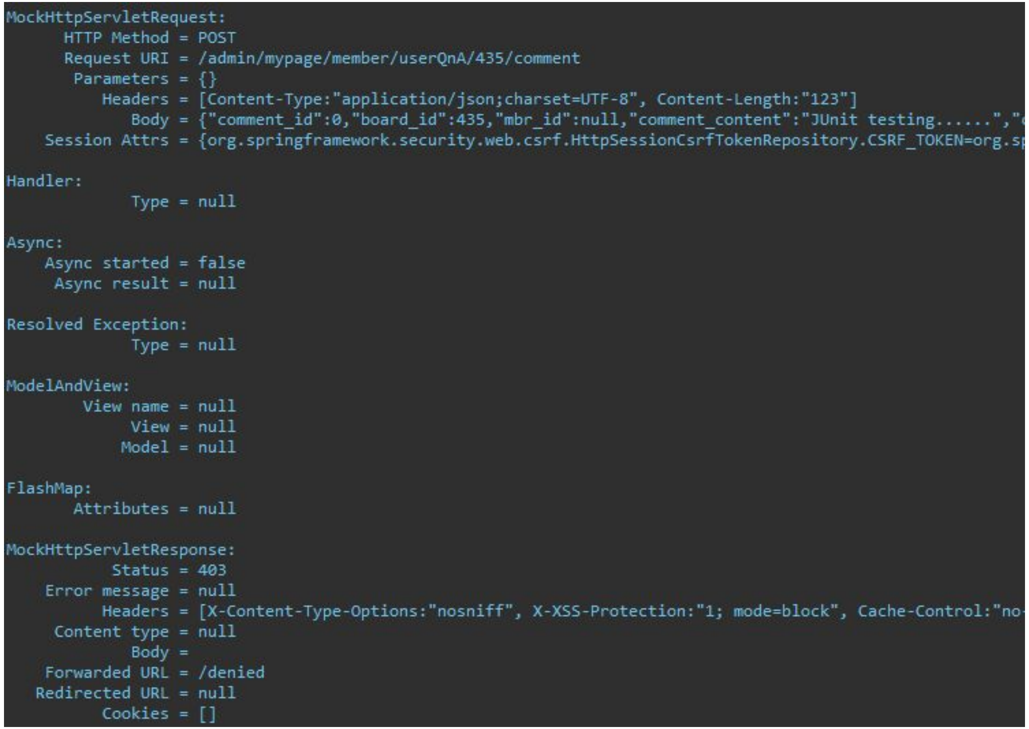
02 Admin 통합 테스트



```
@Before // @Test 이전에 실행
public void setupForadminQnA_commentTest() {
    this.mockMvc = MockMvcBuilders.standaloneSetup(adminController) //
            .build(); //
    mapper = new ObjectMapper();
}

@Test
@WithUserDetails("admin")
public void adminQnA_commentTest() throws Exception {
    log.info("======adminQnA_commentTest======");
    BoardCommentVO comment = new BoardCommentVO();
    comment.setBoard_id(435);
    comment.setComment_content("JUnit testing......");
    String content = mapper.writeValueAsString(comment);

    mockMvc.perform(post("/admin/mypage/member/userQnA/435/comment") // postmapping test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}
```

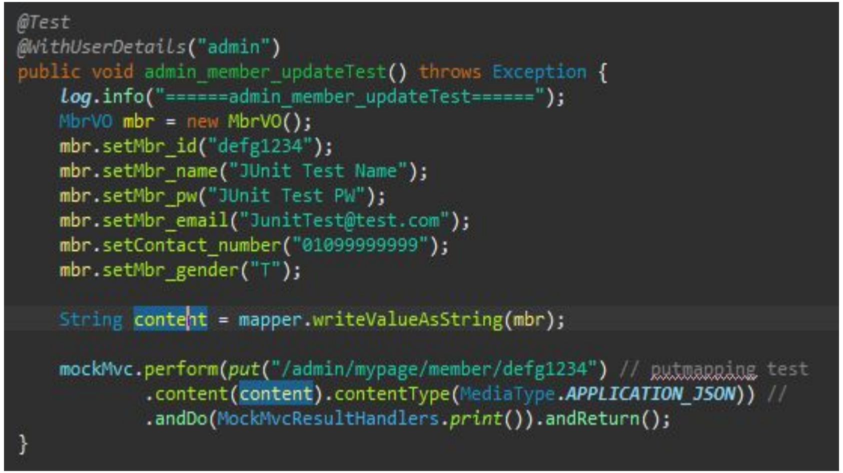


```
Finished after 6.823 seconds

Runs: 3/3        Errors: 0        Failures: 0

AdminControllerTest [Runner: JUnit 5] (2.073 s)
    admin_seller_listTest() (1.898 s)
    admin_member_updateTest() (0.132 s)
    adminQnA_commentTest() (0.043 s)
```

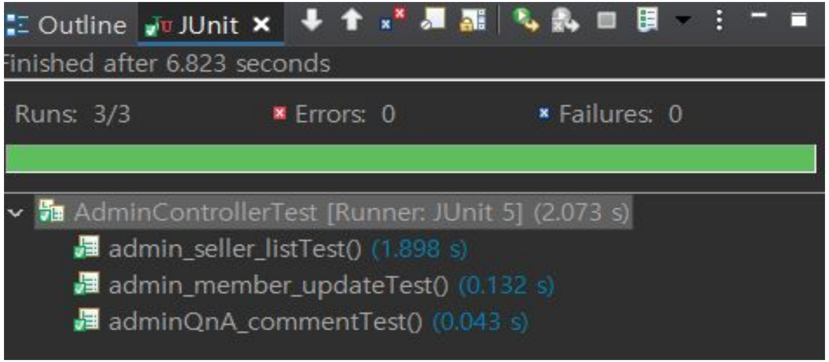


```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /admin/mypage/member/userQnA/435/comment
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"123"]
             Body = {"comment_id":0,"board_id":435,"mbr_id":null,"comment_content":"JUnit testing......","
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.sp

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
          Cookies = []
```

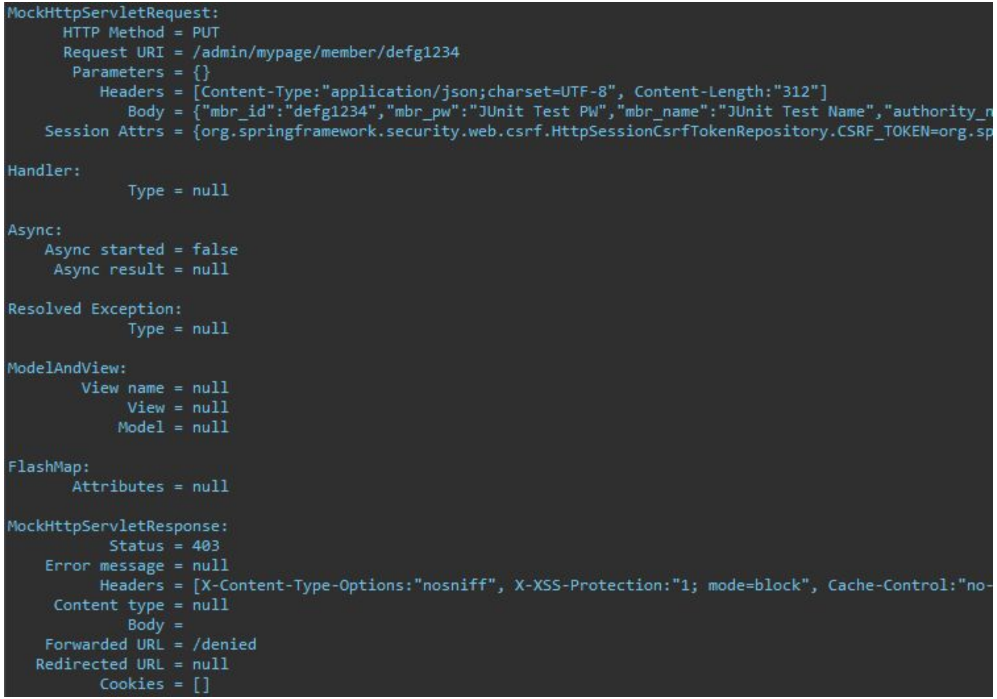
02 Admin 통합 테스트



```java
@Test
@WithUserDetails("admin")
public void admin_member_updateTest() throws Exception {
    log.info("======admin_member_updateTest======");
    MbrVO mbr = new MbrVO();
    mbr.setMbr_id("defg1234");
    mbr.setMbr_name("JUnit Test Name");
    mbr.setMbr_pw("JUnit Test PW");
    mbr.setMbr_email("JunitTest@test.com");
    mbr.setContact_number("01099999999");
    mbr.setMbr_gender("T");

    String content = mapper.writeValueAsString(mbr);

    mockMvc.perform(put("/admin/mypage/member/defg1234") // putmapping test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}
```

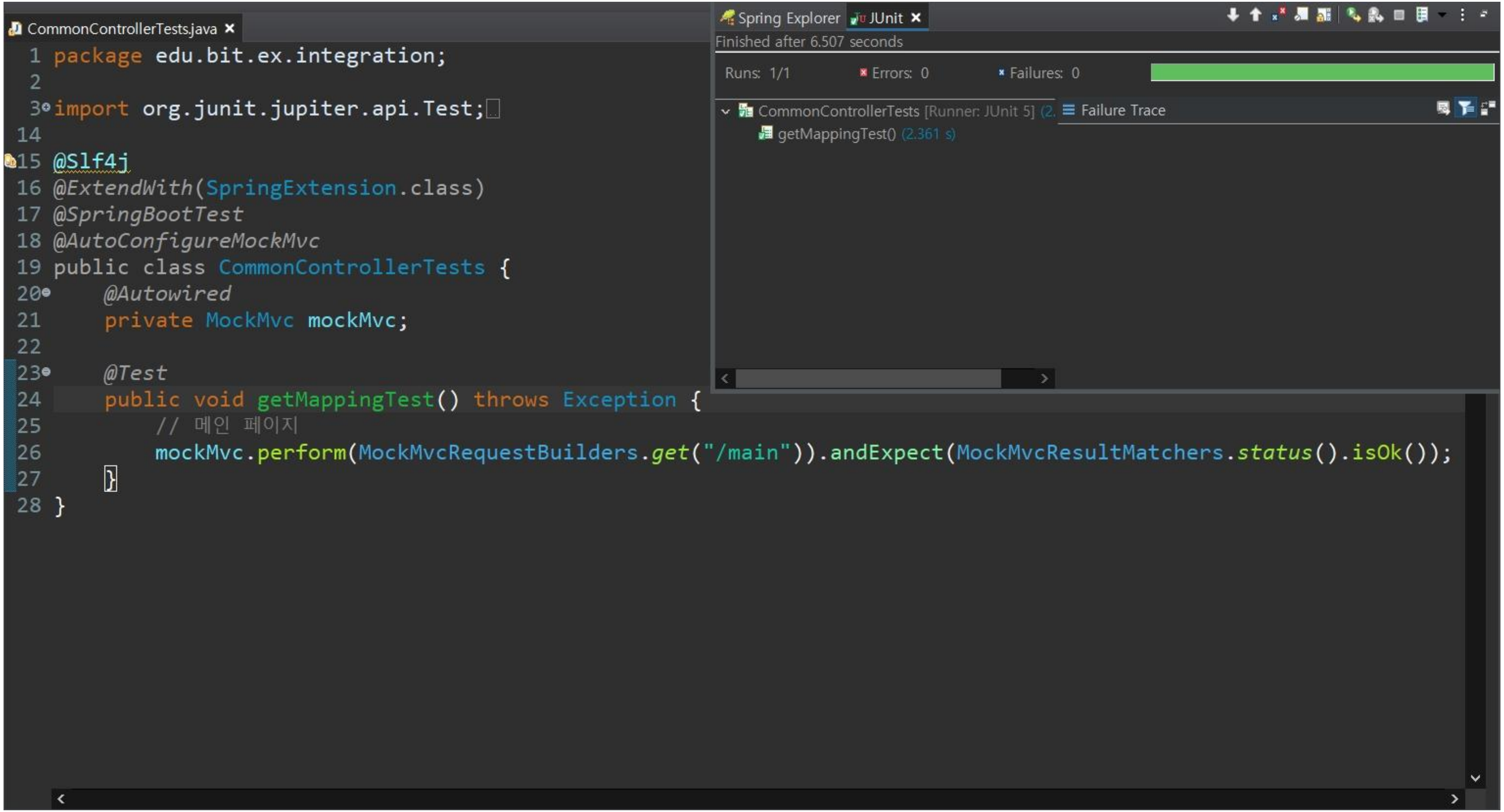


```
Finished after 6.823 seconds

Runs: 3/3        Errors: 0        Failures: 0

AdminControllerTest [Runner: JUnit 5] (2.073 s)
    admin_seller_listTest() (1.898 s)
    admin_member_updateTest() (0.132 s)
    adminQnA_commentTest() (0.043 s)
```



```
MockHttpServletRequest:
      HTTP Method = PUT
      Request URI = /admin/mypage/member/defg1234
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"312"]
             Body = {"mbr_id":"defg1234","mbr_pw":"JUnit Test PW","mbr_name":"JUnit Test Name","authority_n
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.sp

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
          Cookies = []
```

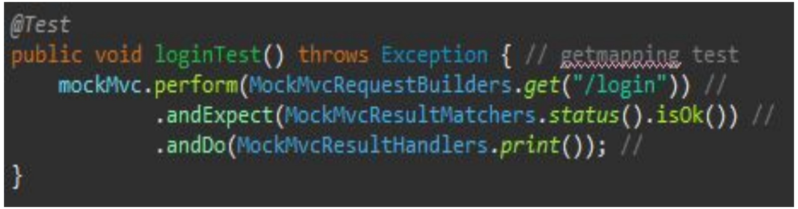
03 Common 통합 테스트



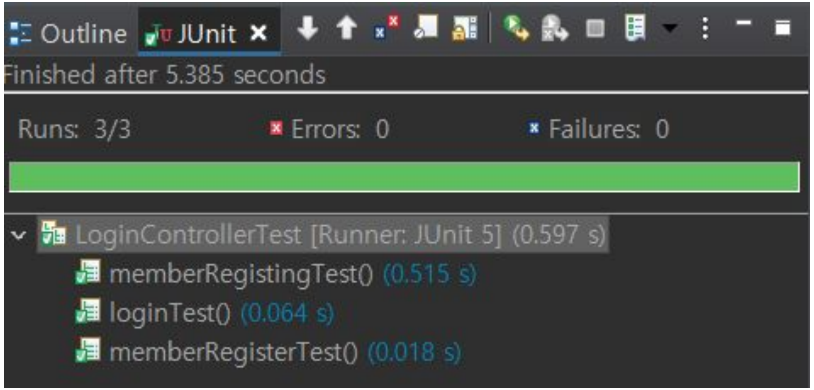
```java
package edu.bit.ex.integration;

import org.junit.jupiter.api.Test;

@Slf4j
@ExtendWith(SpringExtension.class)
@SpringBootTest
@AutoConfigureMockMvc
public class CommonControllerTests {
    @Autowired
    private MockMvc mockMvc;

    @Test
    public void getMappingTest() throws Exception {
        // 메인 페이지
        mockMvc.perform(MockMvcRequestBuilders.get("/main")).andExpect(MockMvcResultMatchers.status().isOk());
    }
}
```

```
Finished after 6.507 seconds

Runs: 1/1        Errors: 0        Failures: 0

CommonControllerTests [Runner: JUnit 5] (2.
    getMappingTest() (2.361 s)
```

04 Login 통합 테스트



```
@Test
public void loginTest() throws Exception { // getmapping test
    mockMvc.perform(MockMvcRequestBuilders.get("/login")) //
            .andExpect(MockMvcResultMatchers.status().isOk()) //
            .andDo(MockMvcResultHandlers.print()); //
}
```

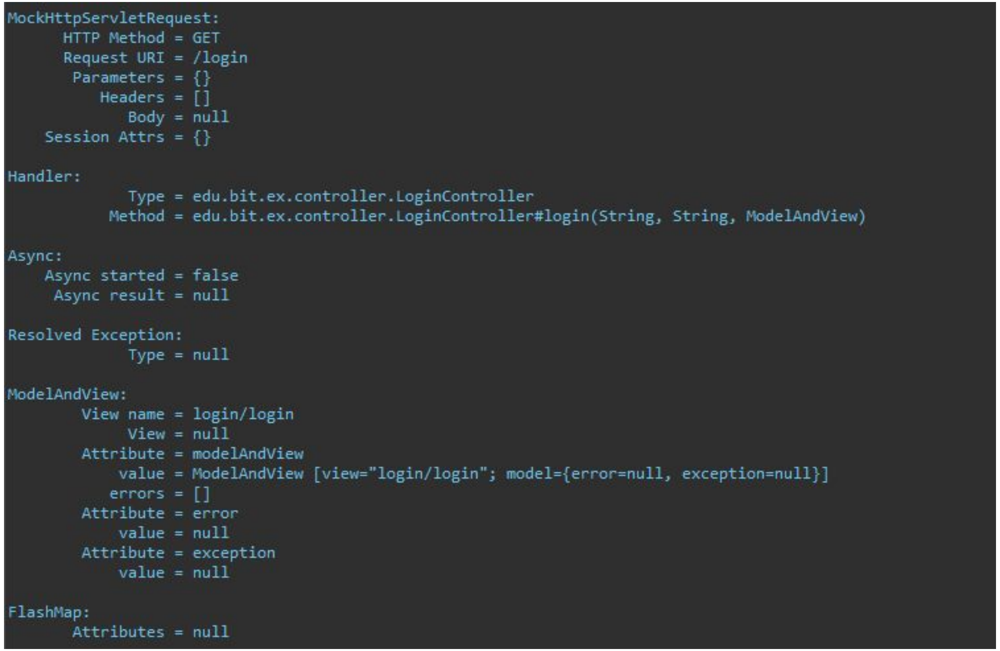


Finished after 5.385 seconds

Runs: 3/3 Errors: 0 Failures: 0

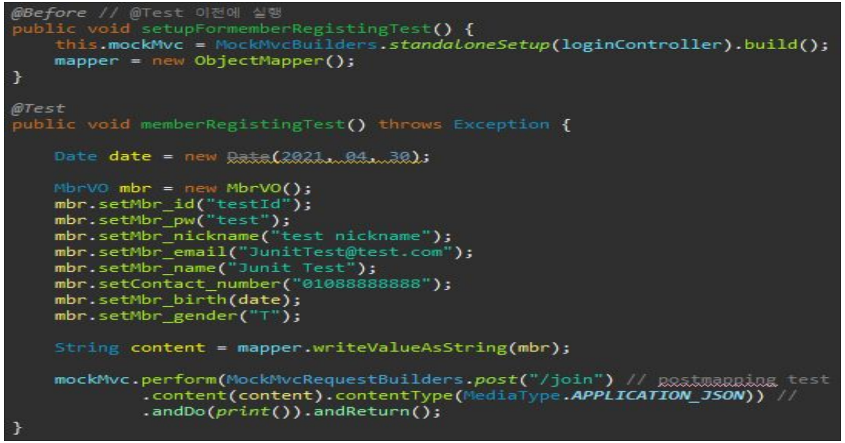
LoginControllerTest [Runner: JUnit 5] (0.597 s)

- memberRegistingTest() (0.515 s)
- loginTest() (0.064 s)
- memberRegisterTest() (0.018 s)

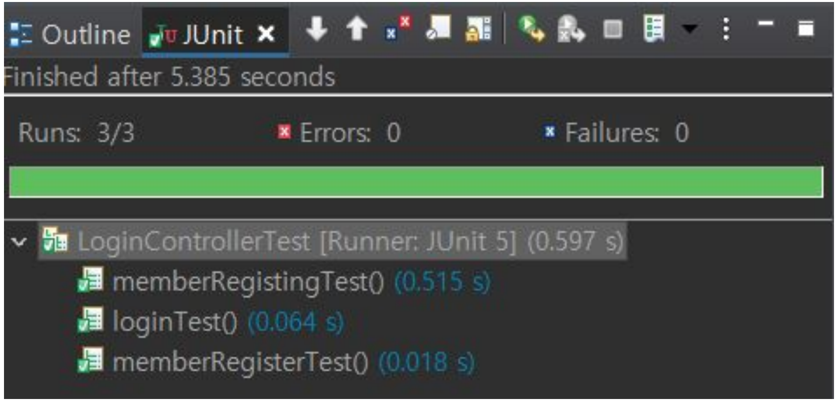


```
MockHttpServletRequest:
      HTTP Method = GET
      Request URI = /login
       Parameters = {}
          Headers = []
             Body = null
    Session Attrs = {}

Handler:
             Type = edu.bit.ex.controller.LoginController
           Method = edu.bit.ex.controller.LoginController#login(String, String, ModelAndView)

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = login/login
             View = null
        Attribute = modelAndView
            value = ModelAndView [view="login/login"; model={error=null, exception=null}]
           errors = []
        Attribute = error
            value = null
        Attribute = exception
            value = null

FlashMap:
       Attributes = null
```

04 Login 통합 테스트



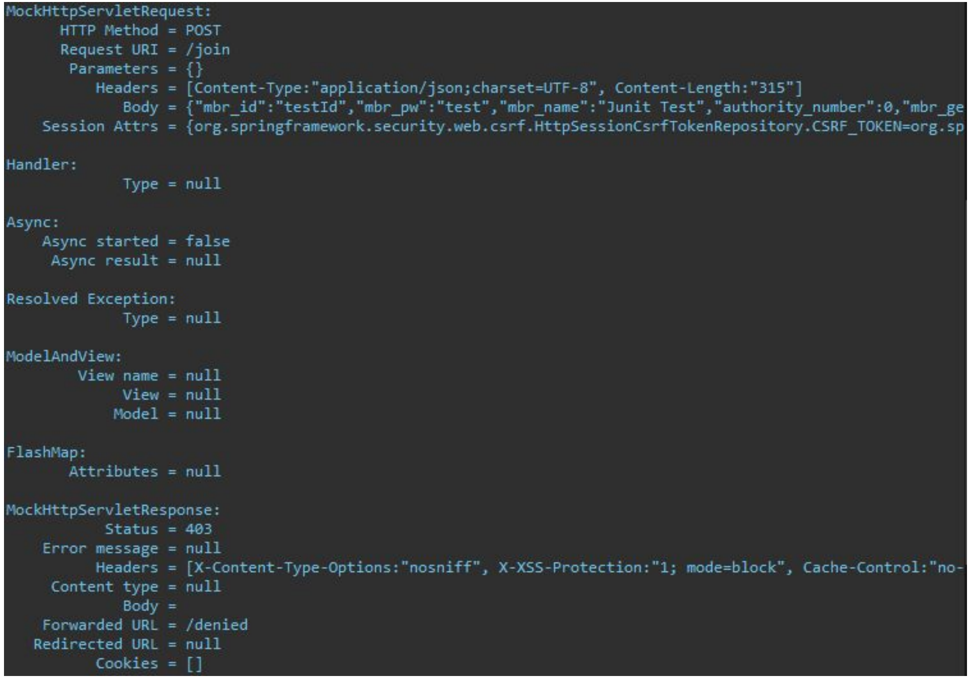
```java
@Before // @Test 이전에 실행
public void setupFormemberRegistingTest() {
    this.mockMvc = MockMvcBuilders.standaloneSetup(loginController).build();
    mapper = new ObjectMapper();
}

@Test
public void memberRegistingTest() throws Exception {

    Date date = new Date(2021, 04, 30);

    MbrVO mbr = new MbrVO();
    mbr.setMbr_id("testId");
    mbr.setMbr_pw("test");
    mbr.setMbr_nickname("test nickname");
    mbr.setMbr_email("JunitTest@test.com");
    mbr.setMbr_name("Junit Test");
    mbr.setContact_number("01088888888");
    mbr.setMbr_birth(date);
    mbr.setMbr_gender("T");

    String content = mapper.writeValueAsString(mbr);

    mockMvc.perform(MockMvcRequestBuilders.post("/join") // postmapping test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(print()).andReturn();
}
```



Finished after 5.385 seconds

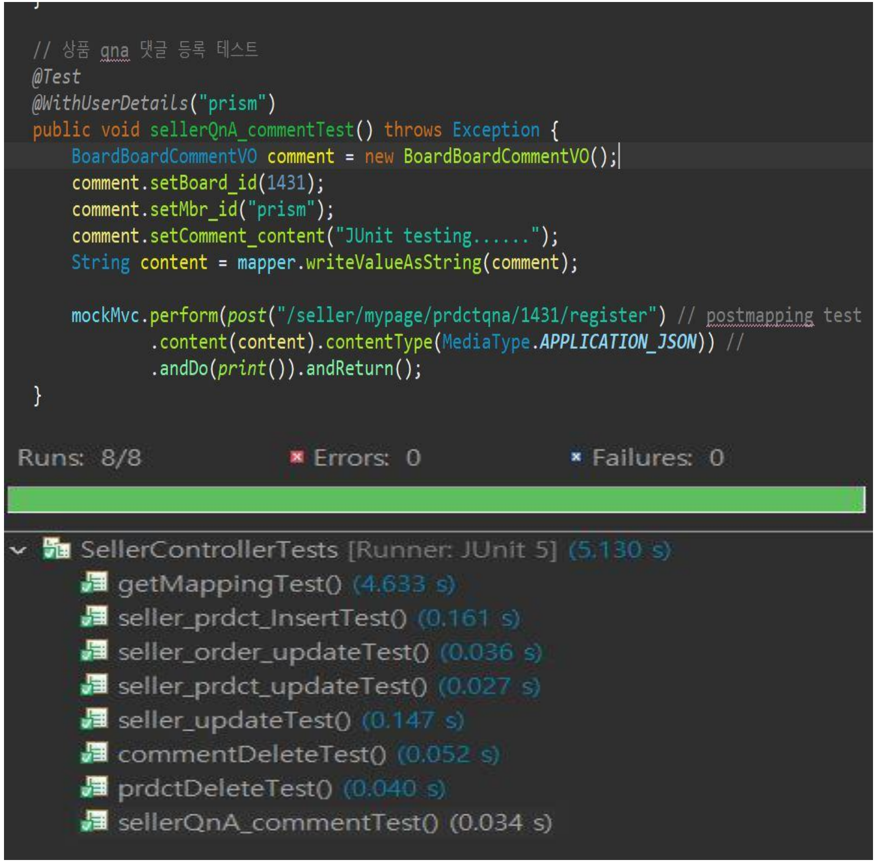
Runs: 3/3 Errors: 0 Failures: 0

- LoginControllerTest [Runner: JUnit 5] (0.597 s)
  - memberRegistingTest() (0.515 s)
  - loginTest() (0.064 s)
  - memberRegisterTest() (0.018 s)



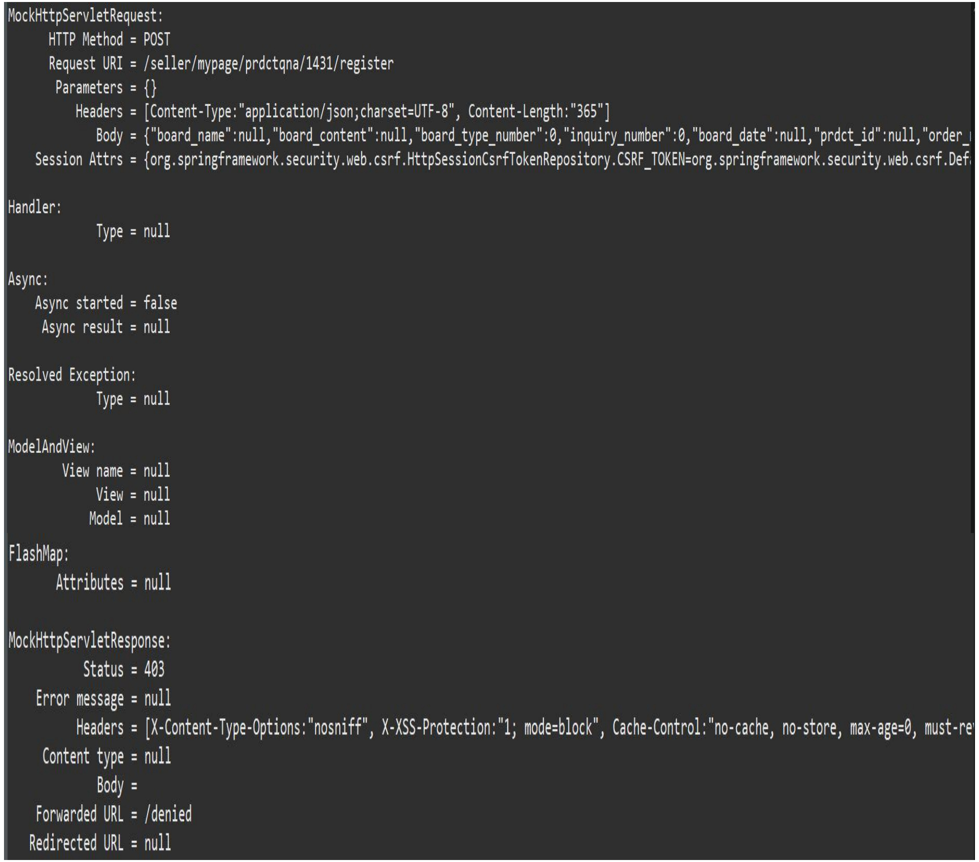
```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /join
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"315"]
             Body = {"mbr_id":"testId","mbr_pw":"test","mbr_name":"Junit Test","authority_number":0,"mbr_ge
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.sp

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
          Cookies = []
```

04 Login 통합 테스트



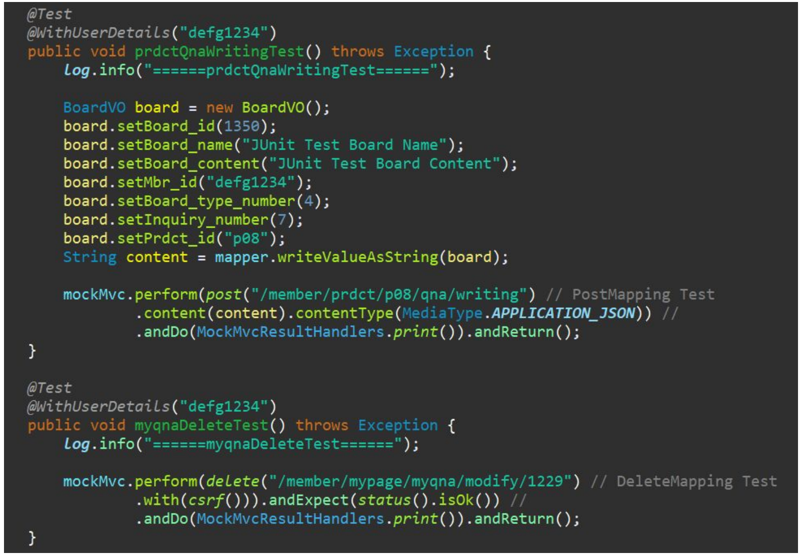
```
// 상품 qna 댓글 등록 테스트
@Test
@WithUserDetails("prism")
public void sellerQnA_commentTest() throws Exception {
    BoardBoardCommentVO comment = new BoardBoardCommentVO();
    comment.setBoard_id(1431);
    comment.setMbr_id("prism");
    comment.setComment_content("JUnit testing......");
    String content = mapper.writeValueAsString(comment);

    mockMvc.perform(post("/seller/mypage/prdctqna/1431/register") // postmapping test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(print()).andReturn();
}
```

```
Runs: 8/8          Errors: 0          Failures: 0

SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```

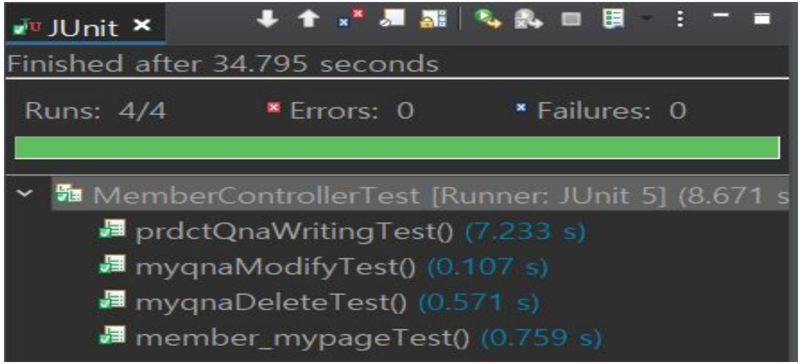


```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /seller/mypage/prdctqna/1431/register
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"365"]
             Body = {"board_name":null,"board_content":null,"board_type_number":0,"inquiry_number":0,"board_date":null,"prdct_id":null,"order_
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security.web.csrf.Def

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-re
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
```

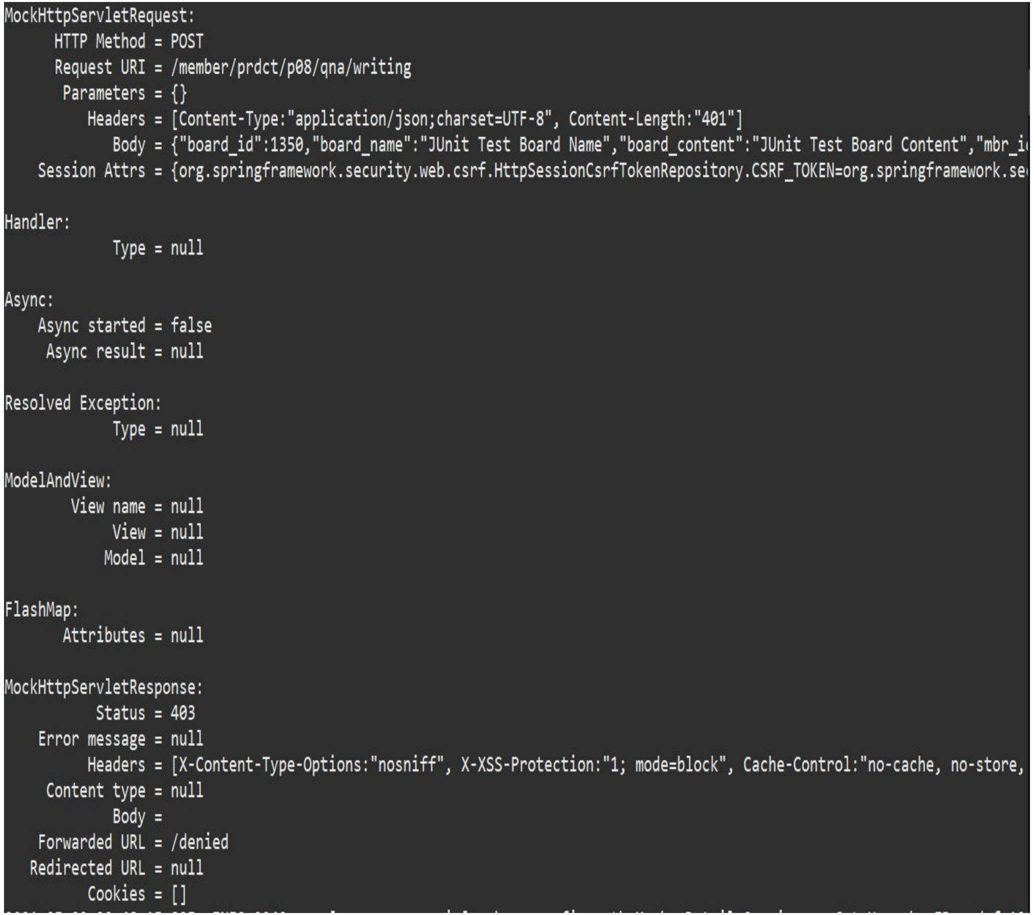
05 Member 통합 테스트



```java
@Test
@WithUserDetails("defg1234")
public void prdctQnaWritingTest() throws Exception {
    log.info("======prdctQnaWritingTest======");

    BoardVO board = new BoardVO();
    board.setBoard_id(1350);
    board.setBoard_name("JUnit Test Board Name");
    board.setBoard_content("JUnit Test Board Content");
    board.setMbr_id("defg1234");
    board.setBoard_type_number(4);
    board.setInquiry_number(7);
    board.setPrdct_id("p08");
    String content = mapper.writeValueAsString(board);

    mockMvc.perform(post("/member/prdct/p08/qna/writing") // PostMapping Test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}

@Test
@WithUserDetails("defg1234")
public void myqnaDeleteTest() throws Exception {
    log.info("======myqnaDeleteTest======");

    mockMvc.perform(delete("/member/mypage/myqna/modify/1229") // DeleteMapping Test
            .with(csrf())).andExpect(status().isOk()) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}
```

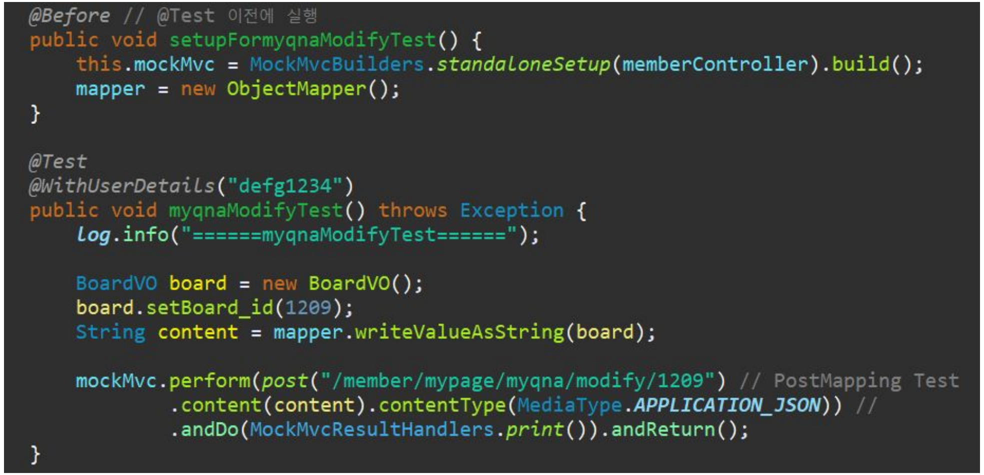


```
Finished after 34.795 seconds

Runs:  4/4        Errors:  0        Failures:  0

MemberControllerTest [Runner: JUnit 5] (8.671 s
    prdctQnaWritingTest() (7.233 s)
    myqnaModifyTest() (0.107 s)
    myqnaDeleteTest() (0.571 s)
    member_mypageTest() (0.759 s)
```

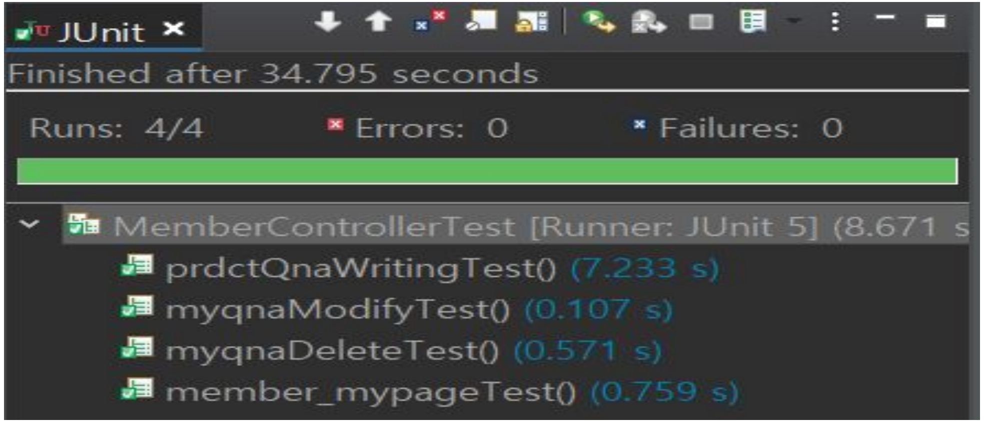


```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /member/prdct/p08/qna/writing
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"401"]
             Body = {"board_id":1350,"board_name":"JUnit Test Board Name","board_content":"JUnit Test Board Content","mbr_i
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.se

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store,
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
          Cookies = []
```

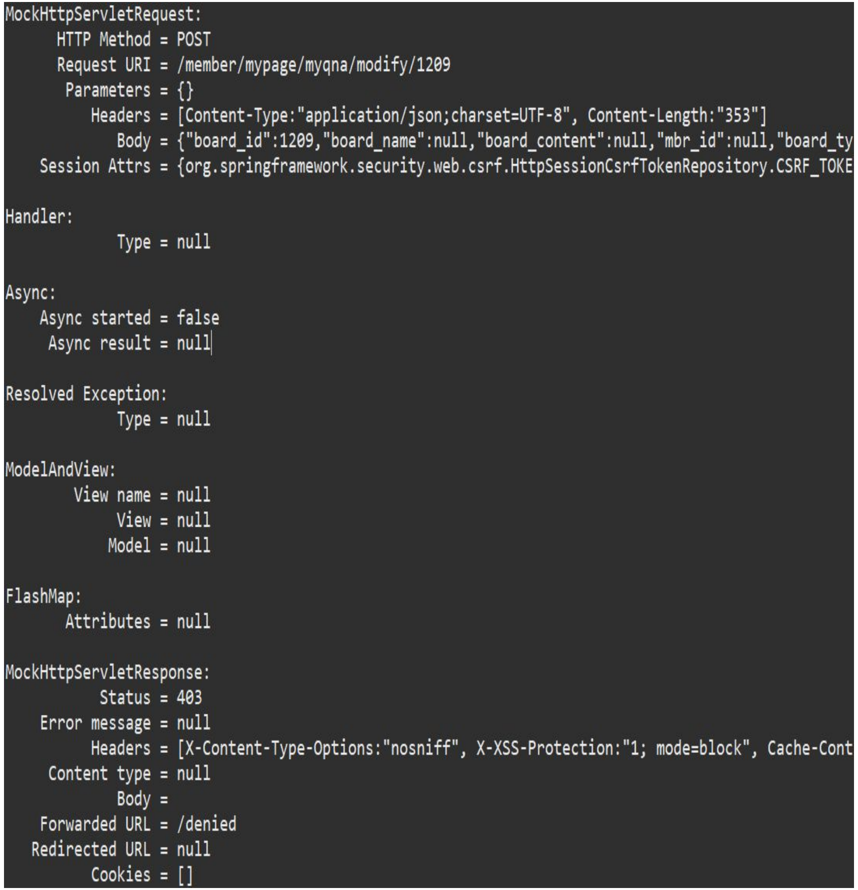
05 Member 통합 테스트



```
@Before // @Test 이전에 실행
public void setupFormyqnaModifyTest() {
    this.mockMvc = MockMvcBuilders.standaloneSetup(memberController).build();
    mapper = new ObjectMapper();
}

@Test
@WithUserDetails("defg1234")
public void myqnaModifyTest() throws Exception {
    log.info("======myqnaModifyTest======");

    BoardVO board = new BoardVO();
    board.setBoard_id(1209);
    String content = mapper.writeValueAsString(board);

    mockMvc.perform(post("/member/mypage/myqna/modify/1209") // PostMapping Test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}
```

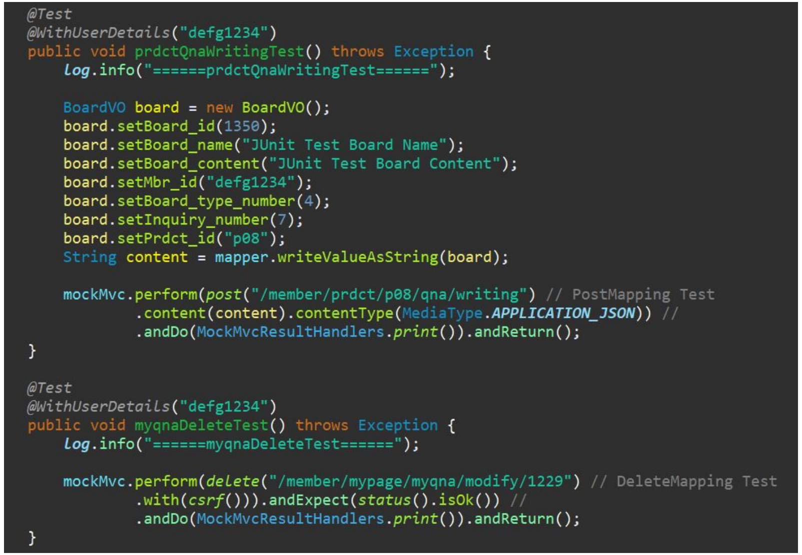


```
Finished after 34.795 seconds

Runs: 4/4        Errors: 0        Failures: 0

MemberControllerTest [Runner: JUnit 5] (8.671 s
    prdctQnaWritingTest() (7.233 s)
    myqnaModifyTest() (0.107 s)
    myqnaDeleteTest() (0.571 s)
    member_mypageTest() (0.759 s)
```

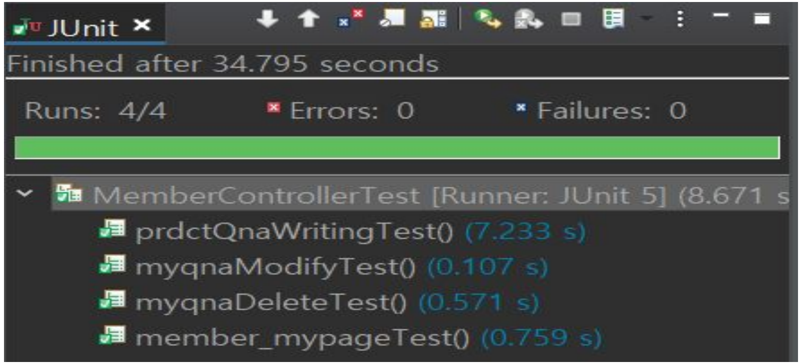


```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /member/mypage/myqna/modify/1209
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"353"]
             Body = {"board_id":1209,"board_name":null,"board_content":null,"mbr_id":null,"board_ty
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKE

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Cont
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
          Cookies = []
```

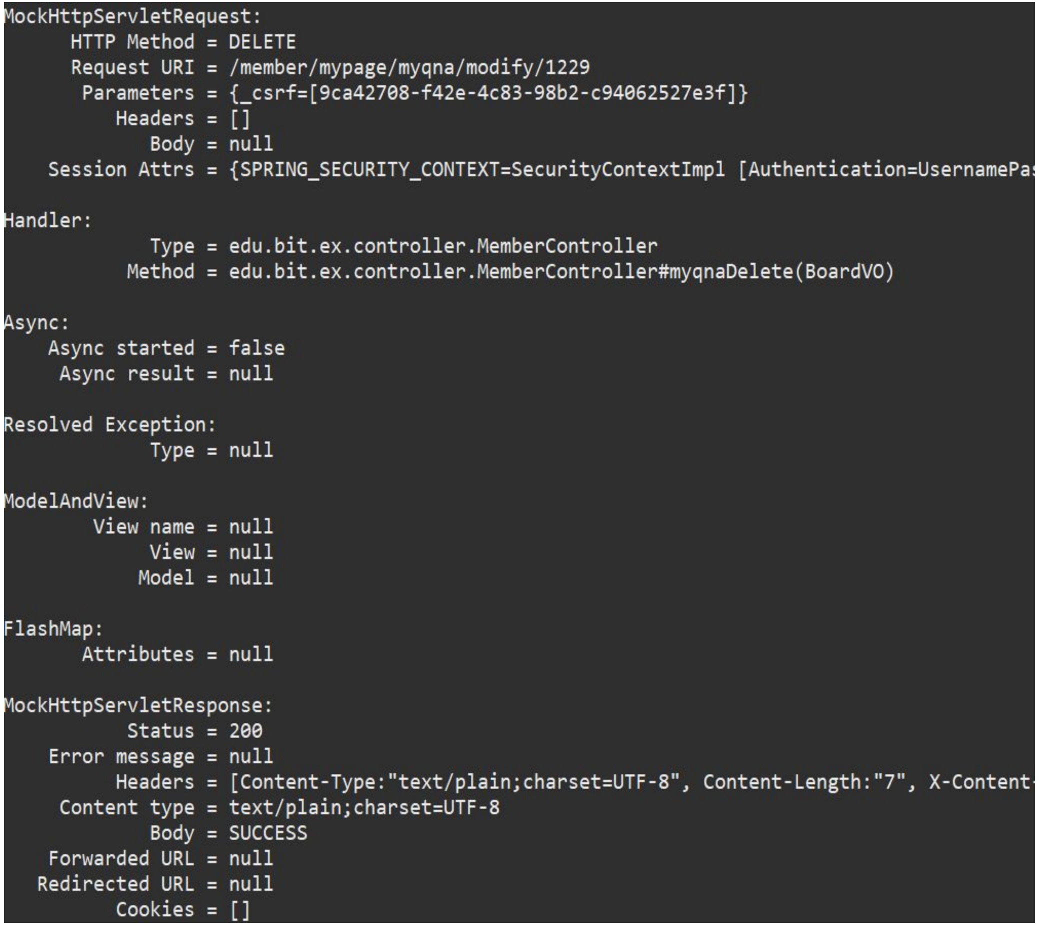
05 Member 통합 테스트



```java
@Test
@WithUserDetails("defg1234")
public void prdctQnaWritingTest() throws Exception {
    log.info("======prdctQnaWritingTest======");

    BoardVO board = new BoardVO();
    board.setBoard_id(1350);
    board.setBoard_name("JUnit Test Board Name");
    board.setBoard_content("JUnit Test Board Content");
    board.setMbr_id("defg1234");
    board.setBoard_type_number(4);
    board.setInquiry_number(7);
    board.setPrdct_id("p08");
    String content = mapper.writeValueAsString(board);

    mockMvc.perform(post("/member/prdct/p08/qna/writing") // PostMapping Test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}

@Test
@WithUserDetails("defg1234")
public void myqnaDeleteTest() throws Exception {
    log.info("======myqnaDeleteTest======");

    mockMvc.perform(delete("/member/mypage/myqna/modify/1229") // DeleteMapping Test
            .with(csrf())).andExpect(status().isOk()) //
            .andDo(MockMvcResultHandlers.print()).andReturn();
}
```

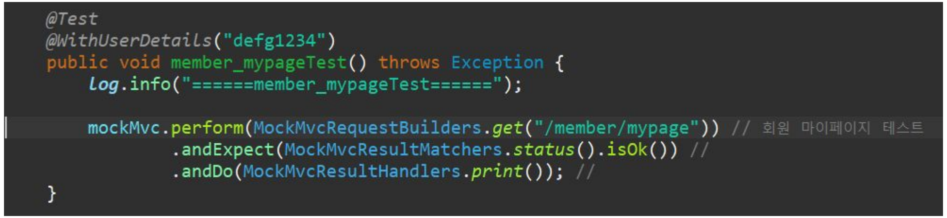


```
Finished after 34.795 seconds

Runs: 4/4        ✖ Errors: 0        ✖ Failures: 0

MemberControllerTest [Runner: JUnit 5] (8.671 s
    prdctQnaWritingTest() (7.233 s)
    myqnaModifyTest() (0.107 s)
    myqnaDeleteTest() (0.571 s)
    member_mypageTest() (0.759 s)
```

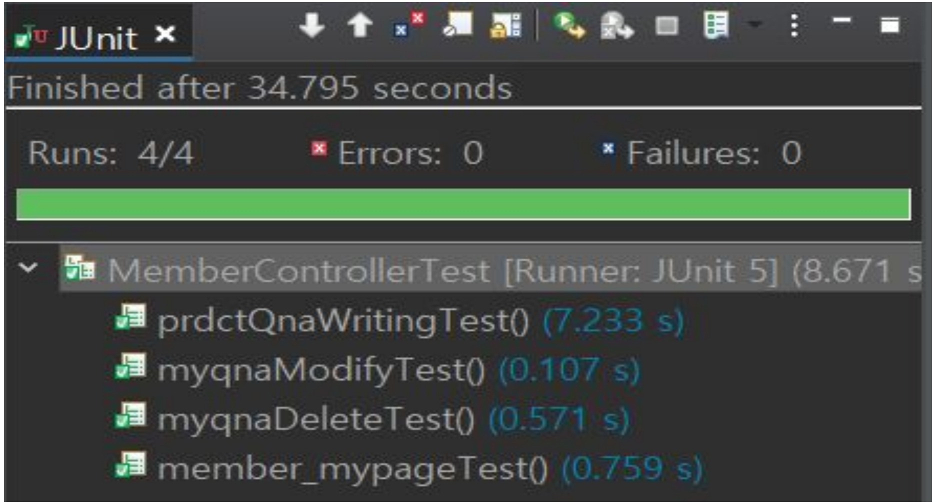


```
MockHttpServletRequest:
      HTTP Method = DELETE
      Request URI = /member/mypage/myqna/modify/1229
       Parameters = {_csrf=[9ca42708-f42e-4c83-98b2-c94062527e3f]}
          Headers = []
             Body = null
    Session Attrs = {SPRING_SECURITY_CONTEXT=SecurityContextImpl [Authentication=UsernamePa

Handler:
             Type = edu.bit.ex.controller.MemberController
           Method = edu.bit.ex.controller.MemberController#myqnaDelete(BoardVO)

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 200
    Error message = null
          Headers = [Content-Type:"text/plain;charset=UTF-8", Content-Length:"7", X-Content-
     Content type = text/plain;charset=UTF-8
             Body = SUCCESS
    Forwarded URL = null
   Redirected URL = null
          Cookies = []
```

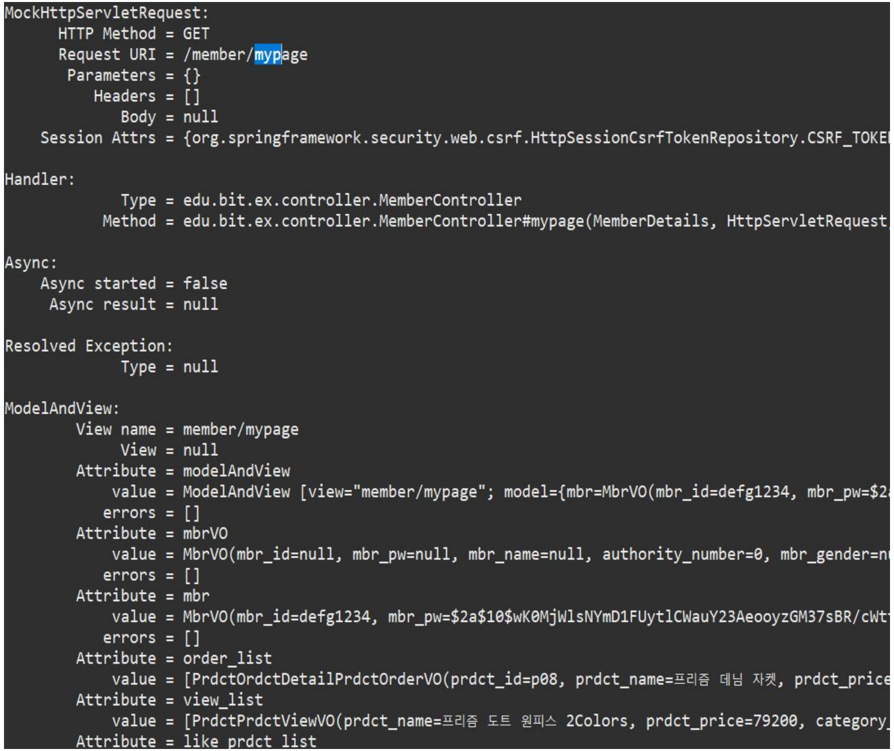
05 Member 통합 테스트



```
@Test
@WithUserDetails("defg1234")
public void member_mypageTest() throws Exception {
    log.info("======member_mypageTest======");

    mockMvc.perform(MockMvcRequestBuilders.get("/member/mypage")) // 회원 마이페이지 테스트
            .andExpect(MockMvcResultMatchers.status().isOk()) //
            .andDo(MockMvcResultHandlers.print()); //
}
```

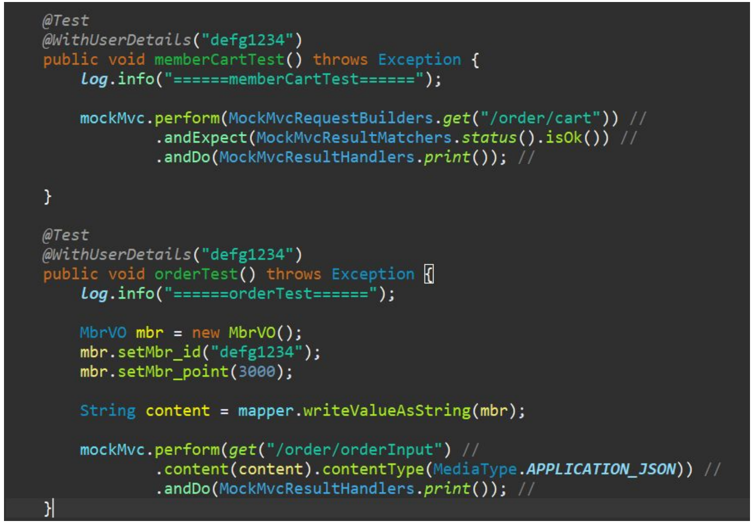


```
Finished after 34.795 seconds

Runs: 4/4        Errors: 0        Failures: 0

MemberControllerTest [Runner: JUnit 5] (8.671 s
    prdctQnaWritingTest() (7.233 s)
    myqnaModifyTest() (0.107 s)
    myqnaDeleteTest() (0.571 s)
    member_mypageTest() (0.759 s)
```

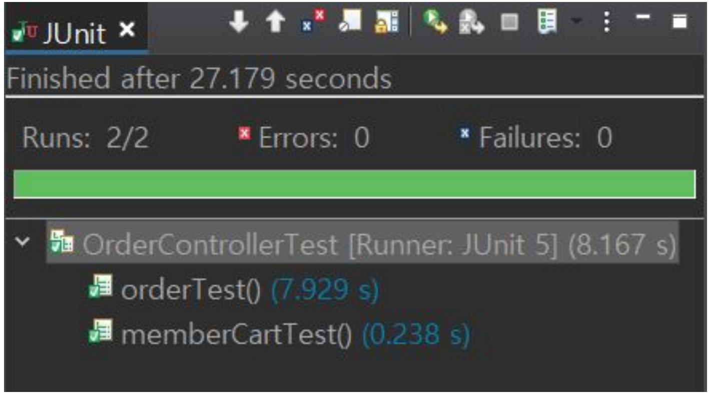


```
MockHttpServletRequest:
      HTTP Method = GET
      Request URI = /member/mypage
       Parameters = {}
          Headers = []
             Body = null
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKE

Handler:
             Type = edu.bit.ex.controller.MemberController
           Method = edu.bit.ex.controller.MemberController#mypage(MemberDetails, HttpServletRequest

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = member/mypage
             View = null
        Attribute = modelAndView
            value = ModelAndView [view="member/mypage"; model={mbr=MbrVO(mbr_id=defg1234, mbr_pw=$2
           errors = []
        Attribute = mbrVO
            value = MbrVO(mbr_id=null, mbr_pw=null, mbr_name=null, authority_number=0, mbr_gender=n
           errors = []
        Attribute = mbr
            value = MbrVO(mbr_id=defg1234, mbr_pw=$2a$10$wK0MjWlsNYmD1FUytlCWauY23AeooyzGM37sBR/cWt
           errors = []
        Attribute = order_list
            value = [PrdctOrdctDetailPrdctOrderVO(prdct_id=p08, prdct_name=프리즘 데님 자켓, prdct_price
        Attribute = view_list
            value = [PrdctPrdctViewVO(prdct_name=프리즘 도트 원피스 2Colors, prdct_price=79200, category_
        Attribute = like_prdct_list
```

06 Order 통합 테스트



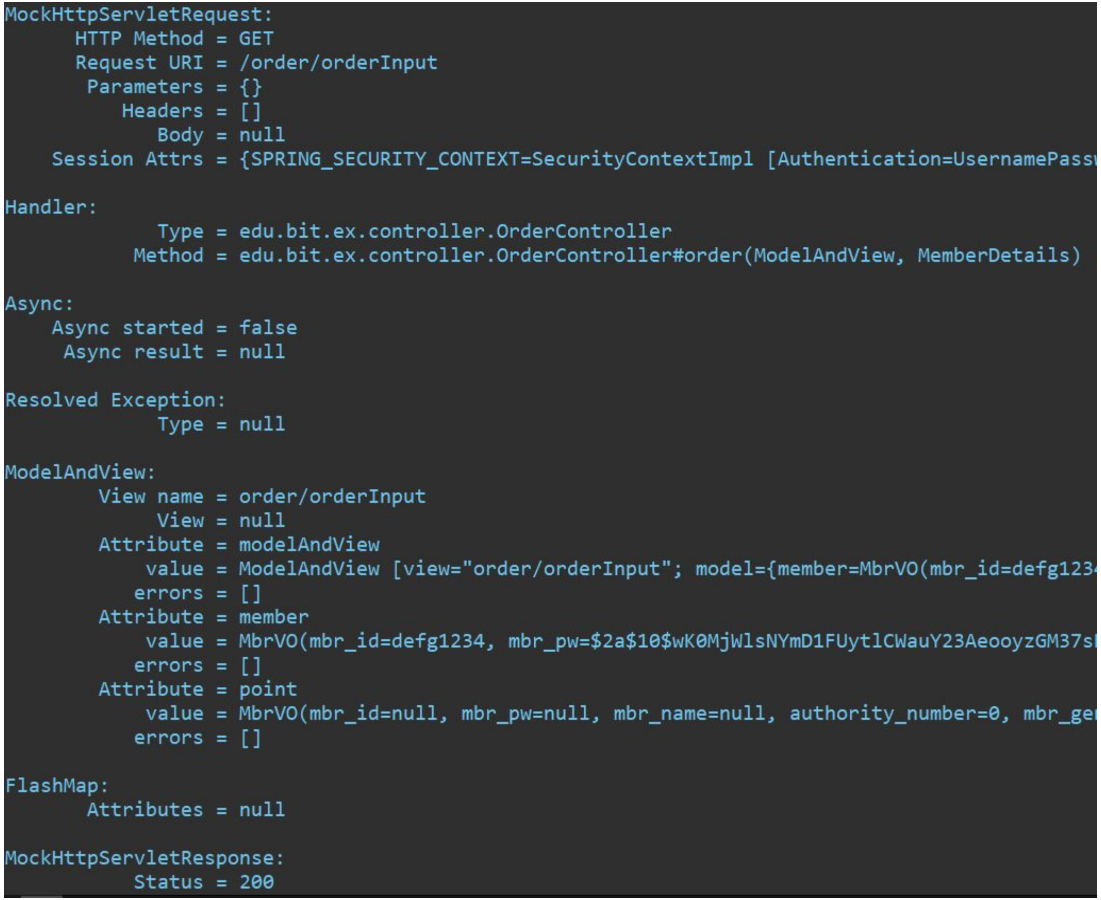
```java
    @Test
    @WithUserDetails("defg1234")
    public void memberCartTest() throws Exception {
        log.info("======memberCartTest======");

        mockMvc.perform(MockMvcRequestBuilders.get("/order/cart")) //
                .andExpect(MockMvcResultMatchers.status().isOk()) //
                .andDo(MockMvcResultHandlers.print()); //

    }

    @Test
    @WithUserDetails("defg1234")
    public void orderTest() throws Exception {
        log.info("======orderTest======");

        MbrVO mbr = new MbrVO();
        mbr.setMbr_id("defg1234");
        mbr.setMbr_point(3000);

        String content = mapper.writeValueAsString(mbr);

        mockMvc.perform(get("/order/orderInput") //
                .content(content).contentType(MediaType.APPLICATION_JSON)) //
                .andDo(MockMvcResultHandlers.print()); //
    }
```



Finished after 27.179 seconds

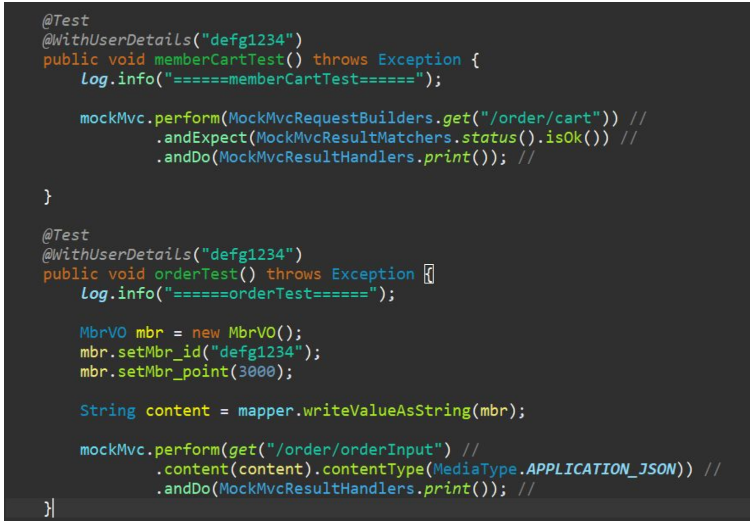
Runs: 2/2   Errors: 0   Failures: 0

- OrderControllerTest [Runner: JUnit 5] (8.167 s)
  - orderTest() (7.929 s)
  - memberCartTest() (0.238 s)

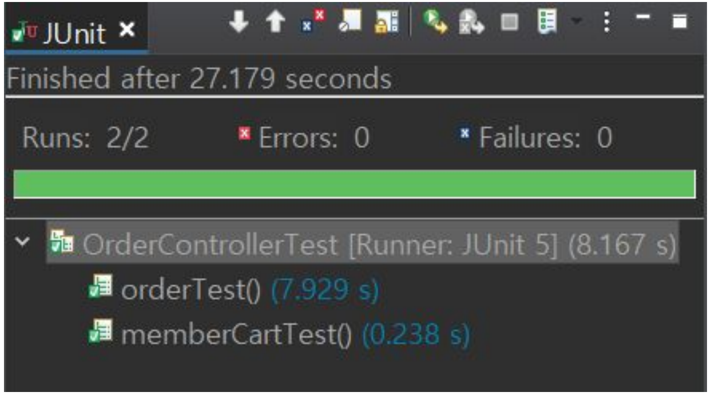


```
MockHttpServletRequest:
      HTTP Method = GET
      Request URI = /order/orderInput
       Parameters = {}
          Headers = []
             Body = null
    Session Attrs = {SPRING_SECURITY_CONTEXT=SecurityContextImpl [Authentication=UsernamePass

Handler:
             Type = edu.bit.ex.controller.OrderController
           Method = edu.bit.ex.controller.OrderController#order(ModelAndView, MemberDetails)

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = order/orderInput
             View = null
        Attribute = modelAndView
            value = ModelAndView [view="order/orderInput"; model={member=MbrVO(mbr_id=defg123
           errors = []
        Attribute = member
            value = MbrVO(mbr_id=defg1234, mbr_pw=$2a$10$wK0MjWlsNYmD1FUytlCWauY23AeooyzGM37s
           errors = []
        Attribute = point
            value = MbrVO(mbr_id=null, mbr_pw=null, mbr_name=null, authority_number=0, mbr_ge
           errors = []

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 200
```

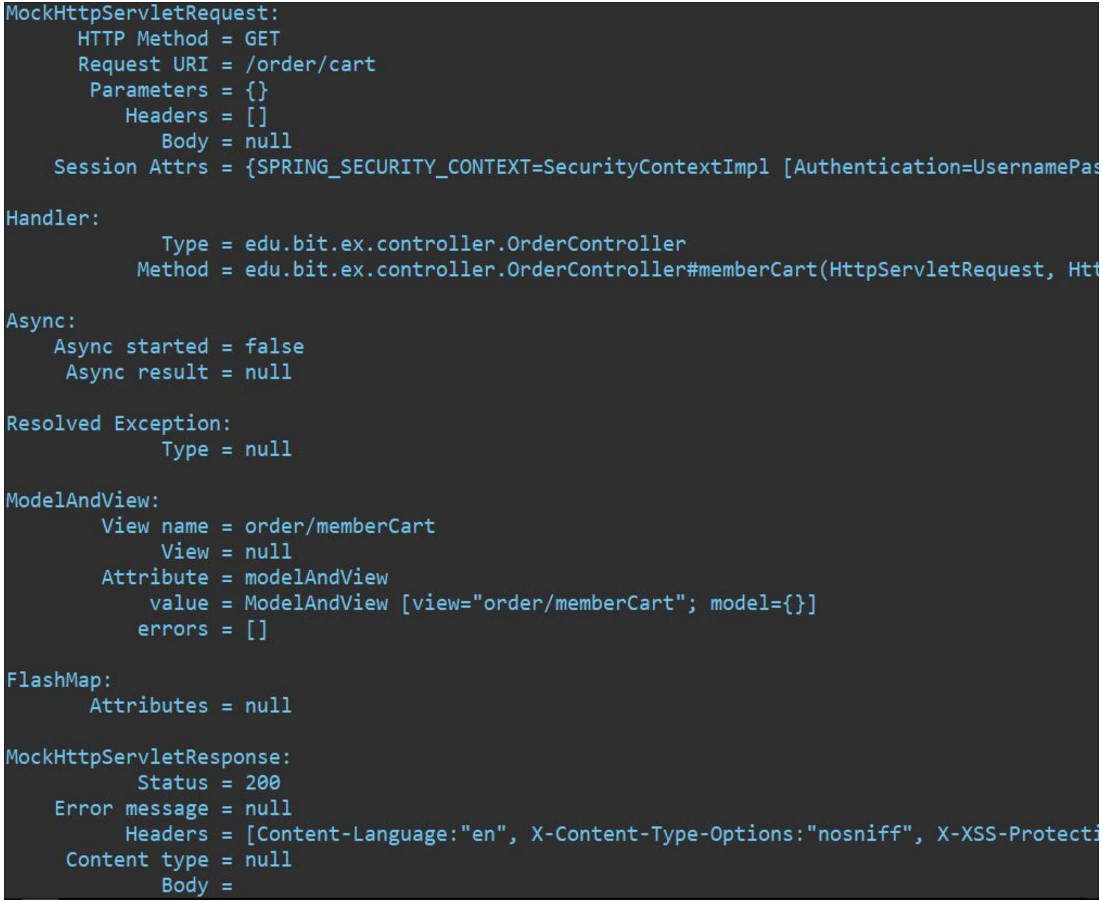
06 Order 통합 테스트



```
    @Test
    @WithUserDetails("defg1234")
    public void memberCartTest() throws Exception {
        log.info("======memberCartTest======");

        mockMvc.perform(MockMvcRequestBuilders.get("/order/cart")) //
                .andExpect(MockMvcResultMatchers.status().isOk()) //
                .andDo(MockMvcResultHandlers.print()); //

    }

    @Test
    @WithUserDetails("defg1234")
    public void orderTest() throws Exception {
        log.info("======orderTest======");

        MbrVO mbr = new MbrVO();
        mbr.setMbr_id("defg1234");
        mbr.setMbr_point(3000);

        String content = mapper.writeValueAsString(mbr);

        mockMvc.perform(get("/order/orderInput") //
                .content(content).contentType(MediaType.APPLICATION_JSON)) //
                .andDo(MockMvcResultHandlers.print()); //
    }
```

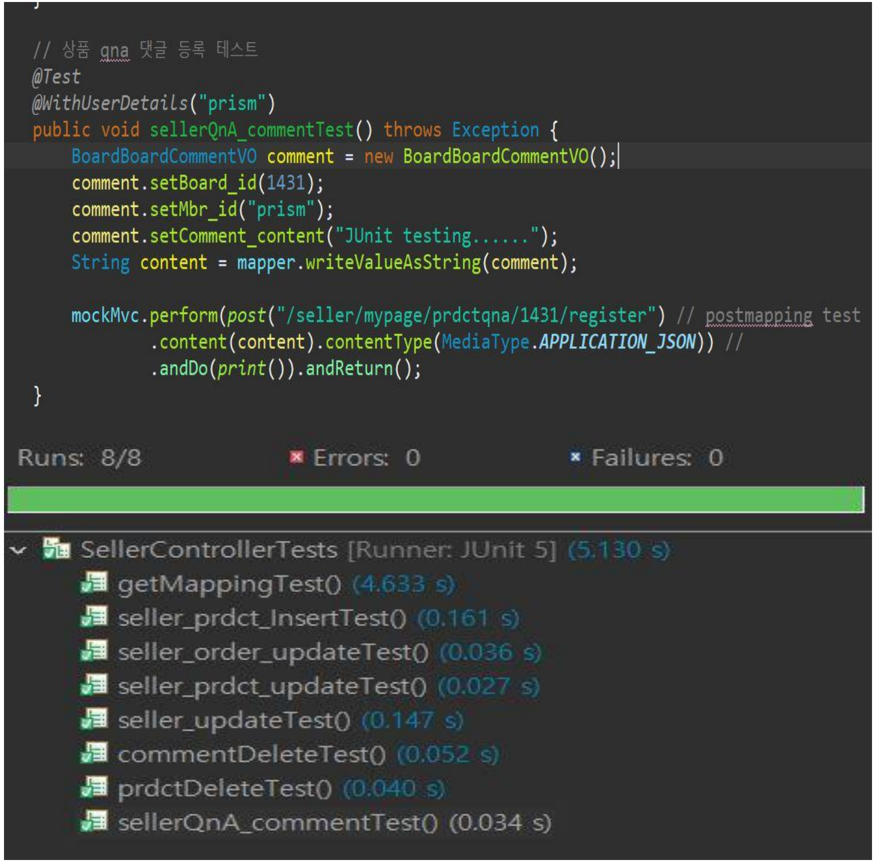


```
Finished after 27.179 seconds

Runs: 2/2        Errors: 0        Failures: 0

OrderControllerTest [Runner: JUnit 5] (8.167 s)
    orderTest() (7.929 s)
    memberCartTest() (0.238 s)
```



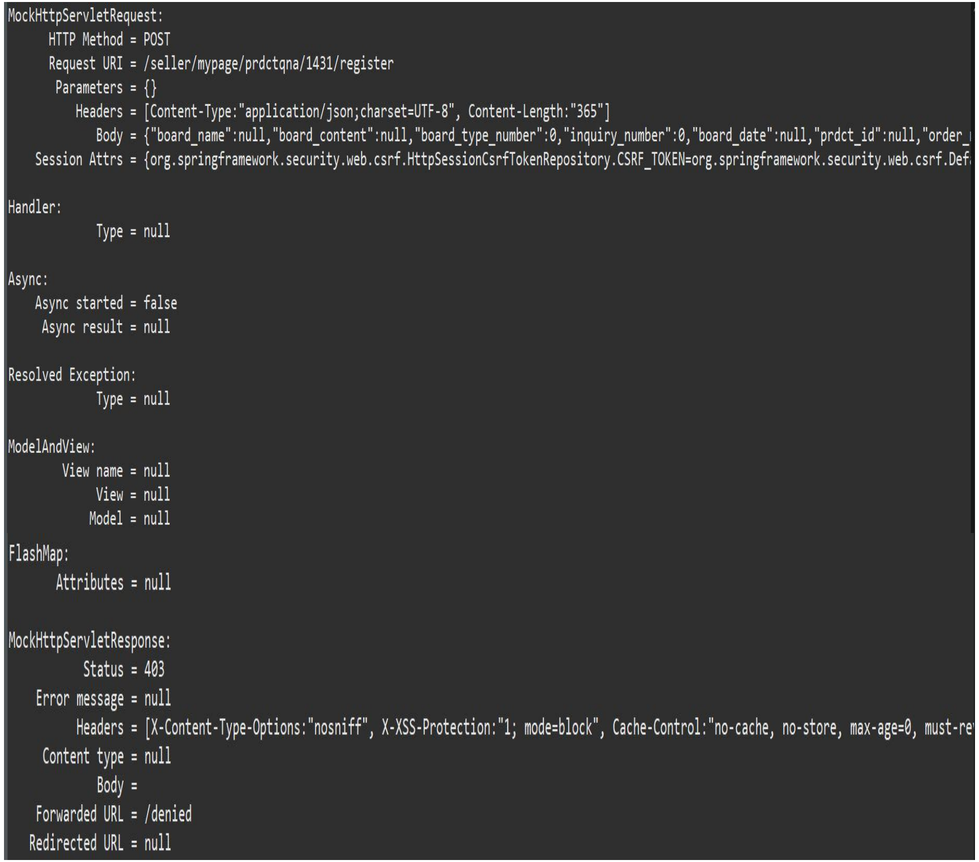
```
MockHttpServletRequest:
      HTTP Method = GET
      Request URI = /order/cart
       Parameters = {}
          Headers = []
             Body = null
    Session Attrs = {SPRING_SECURITY_CONTEXT=SecurityContextImpl [Authentication=UsernamePas

Handler:
             Type = edu.bit.ex.controller.OrderController
           Method = edu.bit.ex.controller.OrderController#memberCart(HttpServletRequest, Htt

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = order/memberCart
             View = null
        Attribute = modelAndView
            value = ModelAndView [view="order/memberCart"; model={}]
           errors = []

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 200
    Error message = null
          Headers = [Content-Language:"en", X-Content-Type-Options:"nosniff", X-XSS-Protecti
     Content type = null
             Body = 
```

07 Seller 통합 테스트



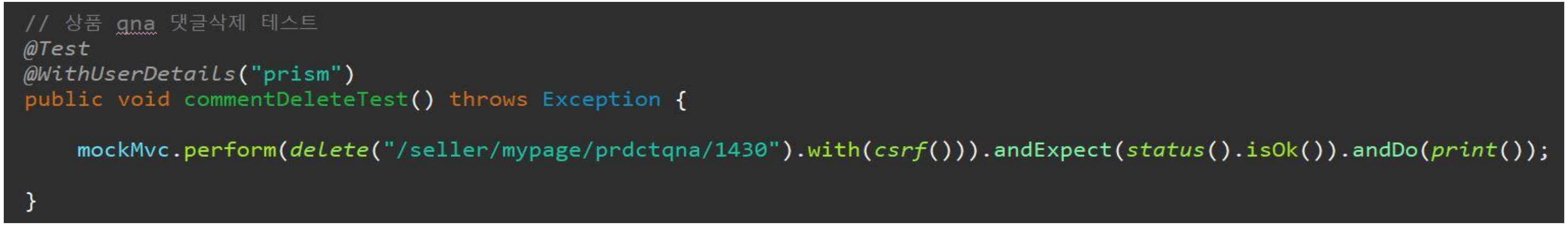
```
// 상품 qna 댓글 등록 테스트
@Test
@WithUserDetails("prism")
public void sellerQnA_commentTest() throws Exception {
    BoardBoardCommentVO comment = new BoardBoardCommentVO();
    comment.setBoard_id(1431);
    comment.setMbr_id("prism");
    comment.setComment_content("JUnit testing......");
    String content = mapper.writeValueAsString(comment);

    mockMvc.perform(post("/seller/mypage/prdctqna/1431/register") // postmapping test
            .content(content).contentType(MediaType.APPLICATION_JSON)) //
            .andDo(print()).andReturn();
}
```

```
Runs: 8/8        Errors: 0        Failures: 0

SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```

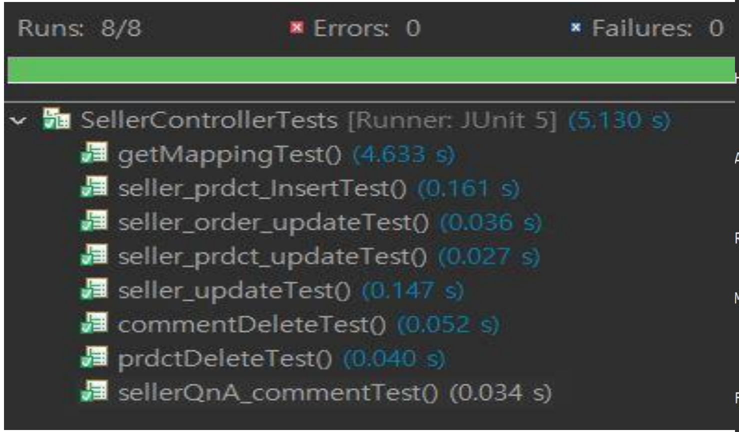


```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /seller/mypage/prdctqna/1431/register
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"365"]
             Body = {"board_name":null,"board_content":null,"board_type_number":0,"inquiry_number":0,"board_date":null,"prdct_id":null,"order_
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security.web.csrf.Def

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-re
     Content type = null
             Body = 
    Forwarded URL = /denied
   Redirected URL = null
```

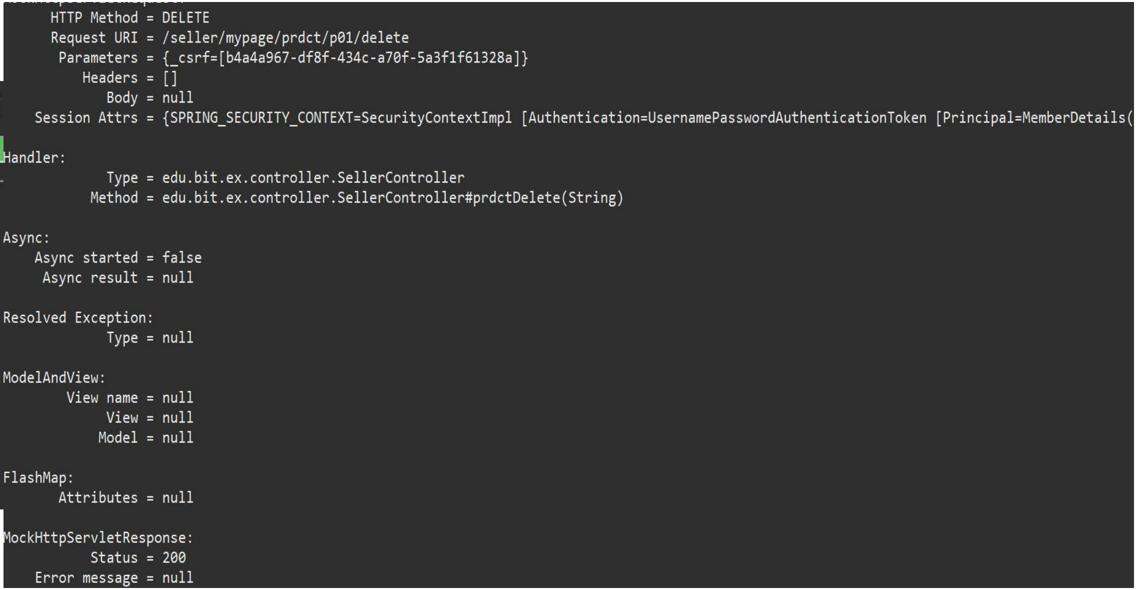
07 Seller 통합 테스트



```java
// 상품 qna 댓글삭제 테스트
@Test
@WithUserDetails("prism")
public void commentDeleteTest() throws Exception {

    mockMvc.perform(delete("/seller/mypage/prdctqna/1430").with(csrf())).andExpect(status().isOk()).andDo(print());

}
```

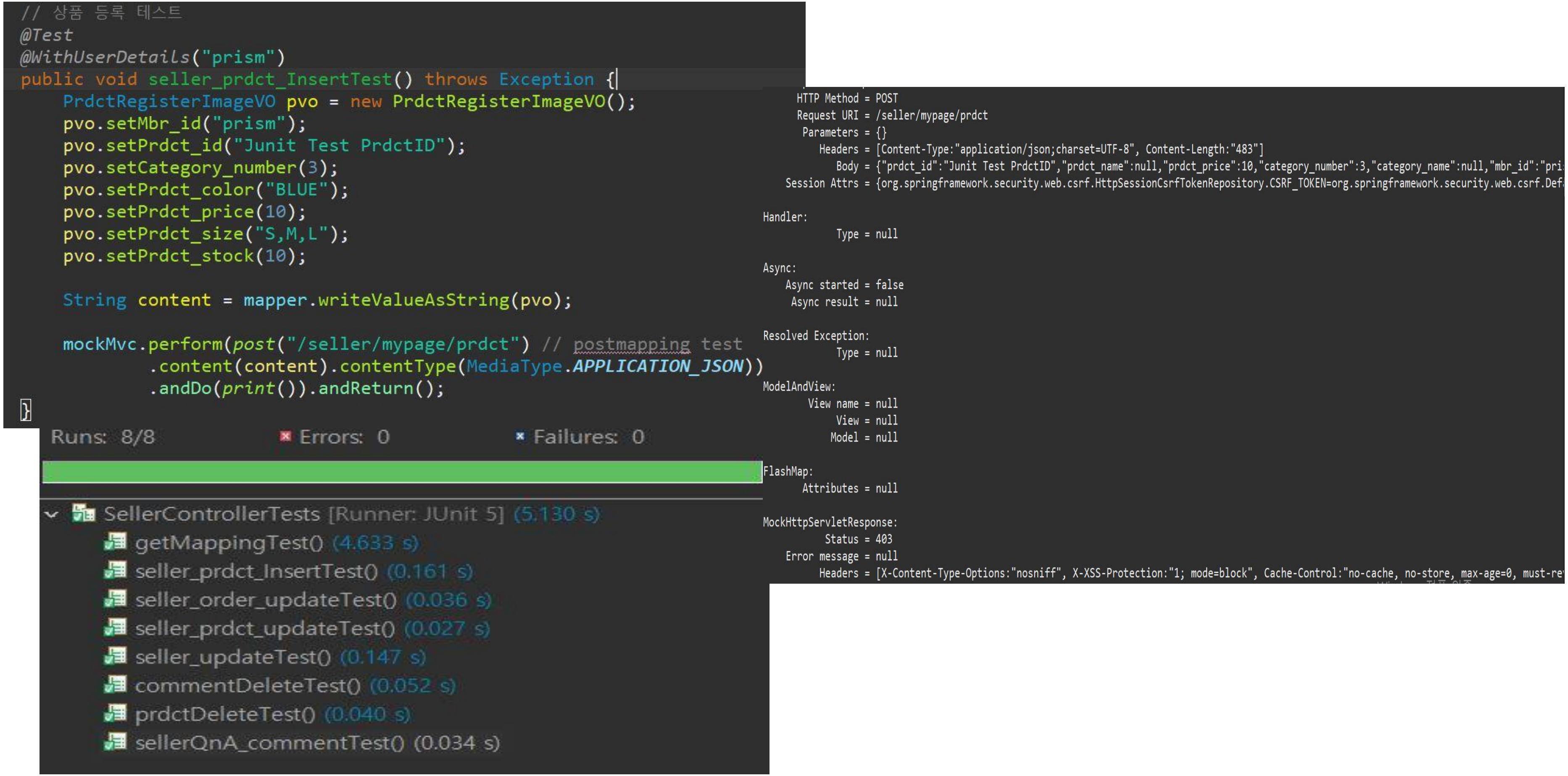


```
Runs: 8/8        ✖ Errors: 0        ✖ Failures: 0

⌄ SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```



```
      HTTP Method = DELETE
      Request URI = /seller/mypage/prdct/p01/delete
       Parameters = {_csrf=[b4a4a967-df8f-434c-a70f-5a3f1f61328a]}
          Headers = []
             Body = null
    Session Attrs = {SPRING_SECURITY_CONTEXT=SecurityContextImpl [Authentication=UsernamePasswordAuthenticationToken [Principal=MemberDetails(

Handler:
             Type = edu.bit.ex.controller.SellerController
           Method = edu.bit.ex.controller.SellerController#prdctDelete(String)

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 200
    Error message = null
```

07 Seller 통합 테스트

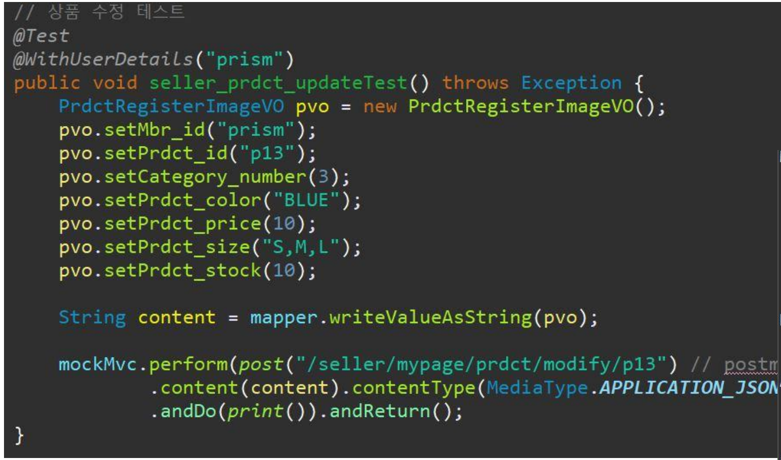


```java
// 상품 등록 테스트
@Test
@WithUserDetails("prism")
public void seller_prdct_InsertTest() throws Exception {
    PrdctRegisterImageVO pvo = new PrdctRegisterImageVO();
    pvo.setMbr_id("prism");
    pvo.setPrdct_id("Junit Test PrdctID");
    pvo.setCategory_number(3);
    pvo.setPrdct_color("BLUE");
    pvo.setPrdct_price(10);
    pvo.setPrdct_size("S,M,L");
    pvo.setPrdct_stock(10);

    String content = mapper.writeValueAsString(pvo);

    mockMvc.perform(post("/seller/mypage/prdct") // postmapping test
            .content(content).contentType(MediaType.APPLICATION_JSON))
            .andDo(print()).andReturn();
}
```

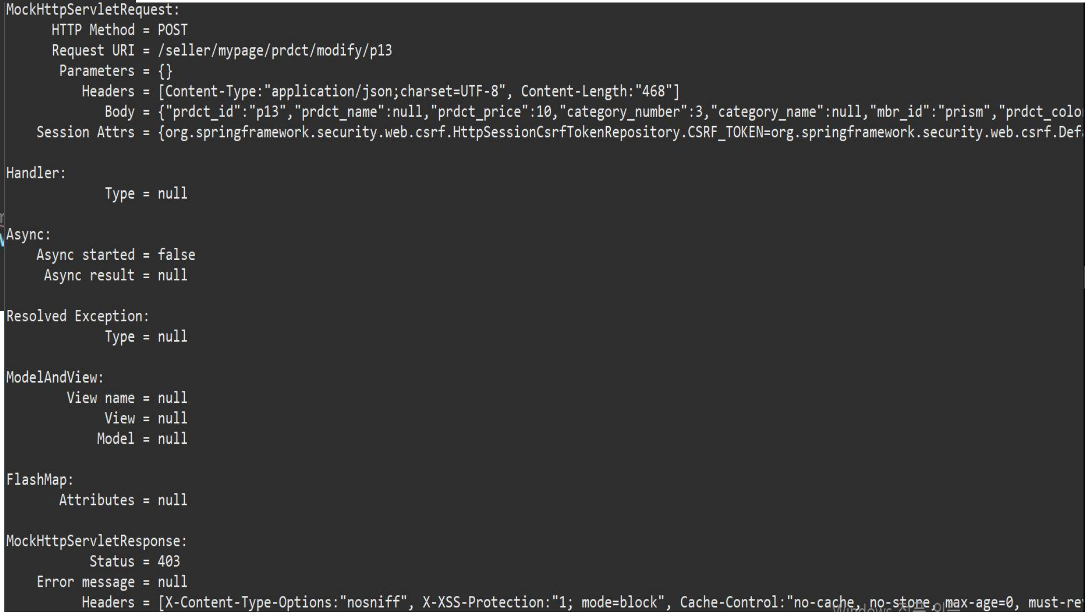
```
      HTTP Method = POST
      Request URI = /seller/mypage/prdct
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"483"]
             Body = {"prdct_id":"Junit Test PrdctID","prdct_name":null,"prdct_price":10,"category_number":3,"category_name":null,"mbr_id":"pri
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security.web.csrf.Def

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-re
```

```
Runs: 8/8        Errors: 0        Failures: 0

SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```

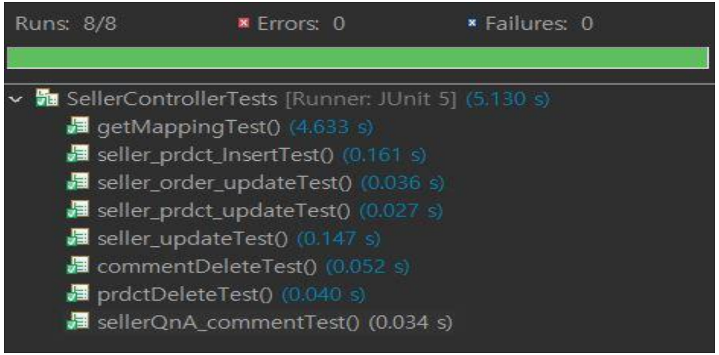
07 Seller 통합 테스트



```java
// 상품 수정 테스트
@Test
@WithUserDetails("prism")
public void seller_prdct_updateTest() throws Exception {
    PrdctRegisterImageVO pvo = new PrdctRegisterImageVO();
    pvo.setMbr_id("prism");
    pvo.setPrdct_id("p13");
    pvo.setCategory_number(3);
    pvo.setPrdct_color("BLUE");
    pvo.setPrdct_price(10);
    pvo.setPrdct_size("S,M,L");
    pvo.setPrdct_stock(10);

    String content = mapper.writeValueAsString(pvo);

    mockMvc.perform(post("/seller/mypage/prdct/modify/p13") // postm
            .content(content).contentType(MediaType.APPLICATION_JSON
            .andDo(print()).andReturn();
}
```

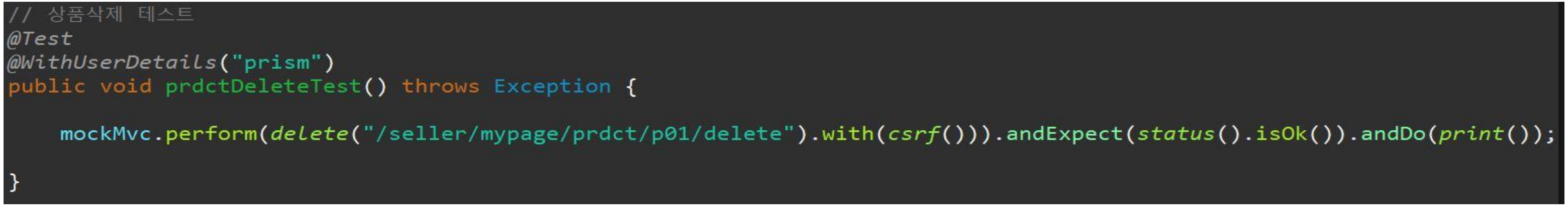


```
MockHttpServletRequest:
      HTTP Method = POST
      Request URI = /seller/mypage/prdct/modify/p13
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"468"]
             Body = {"prdct_id":"p13","prdct_name":null,"prdct_price":10,"category_number":3,"category_name":null,"mbr_id":"prism","prdct_colo
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security.web.csrf.Def

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-re
```

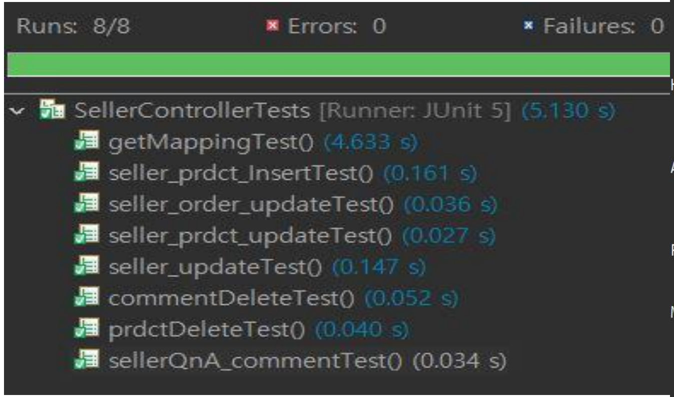


```
Runs: 8/8        Errors: 0        Failures: 0

SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```

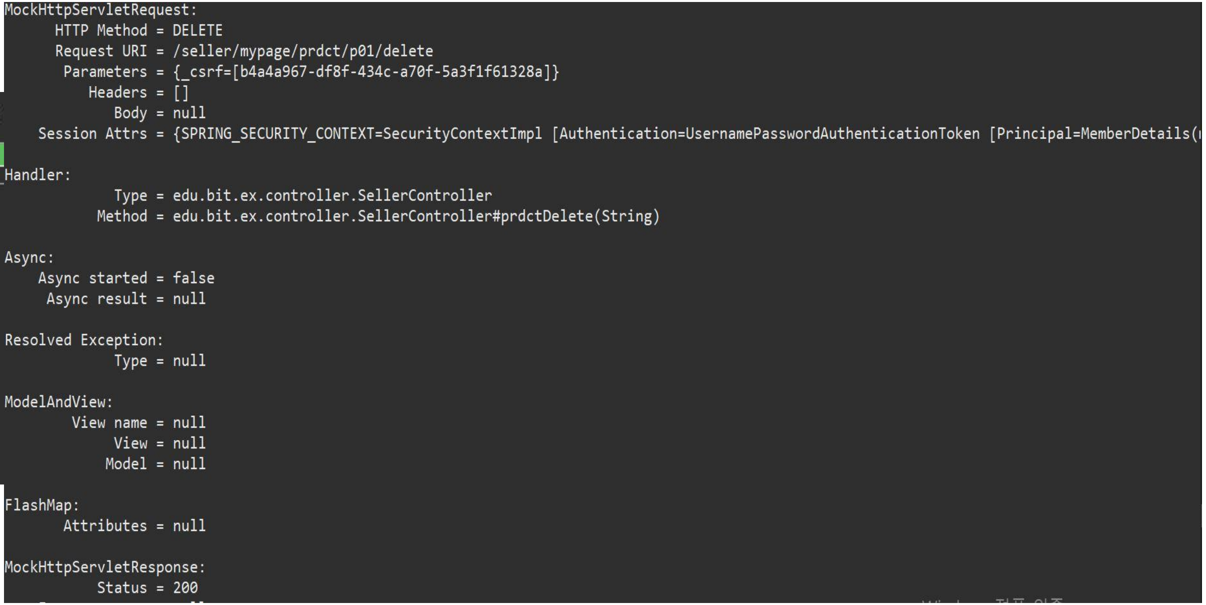
07 Seller 통합 테스트



```
// 상품삭제 테스트
@Test
@WithUserDetails("prism")
public void prdctDeleteTest() throws Exception {

    mockMvc.perform(delete("/seller/mypage/prdct/p01/delete").with(csrf())).andExpect(status().isOk()).andDo(print());

}
```

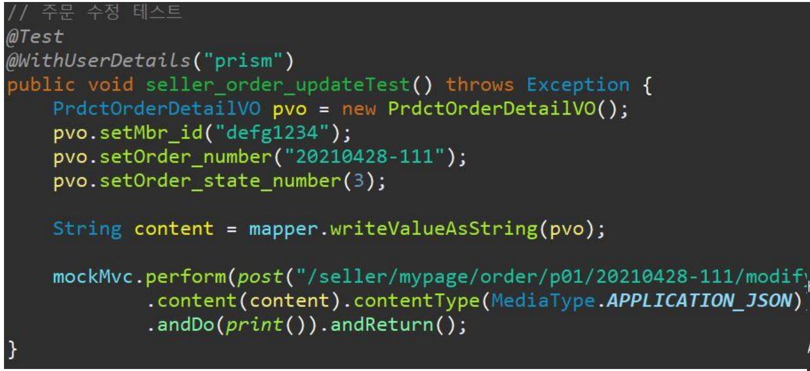


```
Runs: 8/8          ✖ Errors: 0          ✖ Failures: 0

SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```

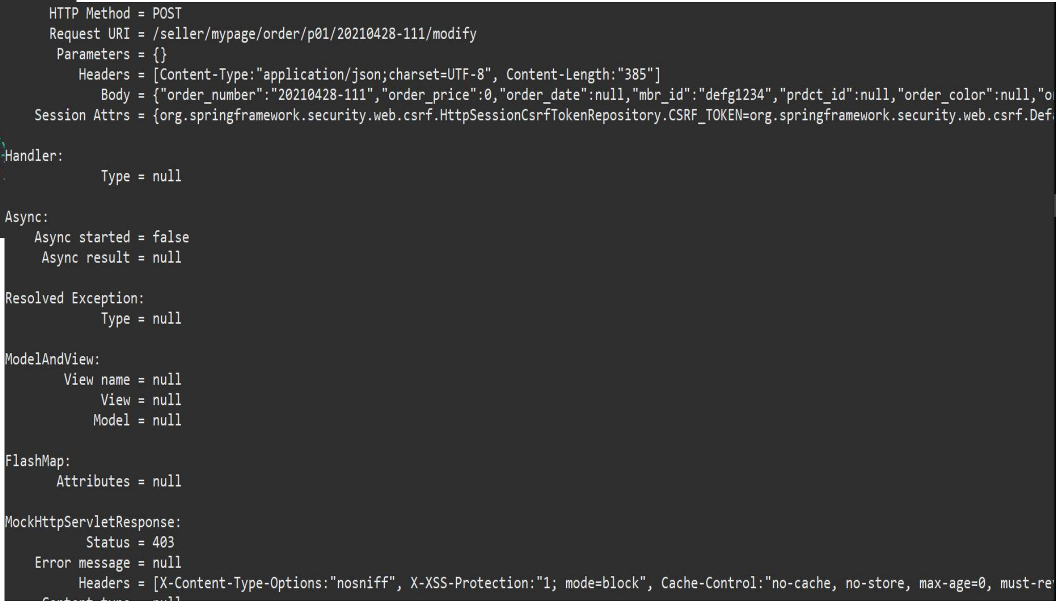


```
MockHttpServletRequest:
      HTTP Method = DELETE
      Request URI = /seller/mypage/prdct/p01/delete
       Parameters = {_csrf=[b4a4a967-df8f-434c-a70f-5a3f1f61328a]}
          Headers = []
             Body = null
    Session Attrs = {SPRING_SECURITY_CONTEXT=SecurityContextImpl [Authentication=UsernamePasswordAuthenticationToken [Principal=MemberDetails(

Handler:
             Type = edu.bit.ex.controller.SellerController
           Method = edu.bit.ex.controller.SellerController#prdctDelete(String)

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 200
```

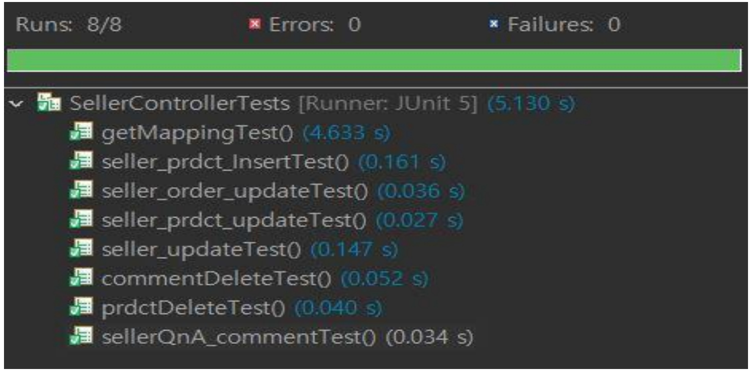
07 Seller 통합 테스트



```
// 주문 수정 테스트
@Test
@WithUserDetails("prism")
public void seller_order_updateTest() throws Exception {
    PrdctOrderDetailVO pvo = new PrdctOrderDetailVO();
    pvo.setMbr_id("defg1234");
    pvo.setOrder_number("20210428-111");
    pvo.setOrder_state_number(3);

    String content = mapper.writeValueAsString(pvo);

    mockMvc.perform(post("/seller/mypage/order/p01/20210428-111/modif
            .content(content).contentType(MediaType.APPLICATION_JSON)
            .andDo(print()).andReturn();
}
```



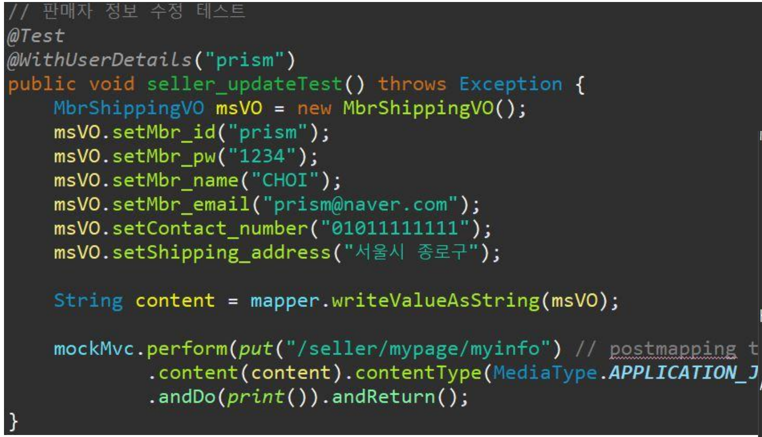
```
   HTTP Method = POST
   Request URI = /seller/mypage/order/p01/20210428-111/modify
    Parameters = {}
       Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"385"]
          Body = {"order_number":"20210428-111","order_price":0,"order_date":null,"mbr_id":"defg1234","prdct_id":null,"order_color":null,"o
 Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security.web.csrf.Def

Handler:
          Type = null

Async:
 Async started = false
  Async result = null

Resolved Exception:
          Type = null

ModelAndView:
     View name = null
          View = null
         Model = null

FlashMap:
    Attributes = null

MockHttpServletResponse:
        Status = 403
 Error message = null
       Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-re
  Content type = null
```



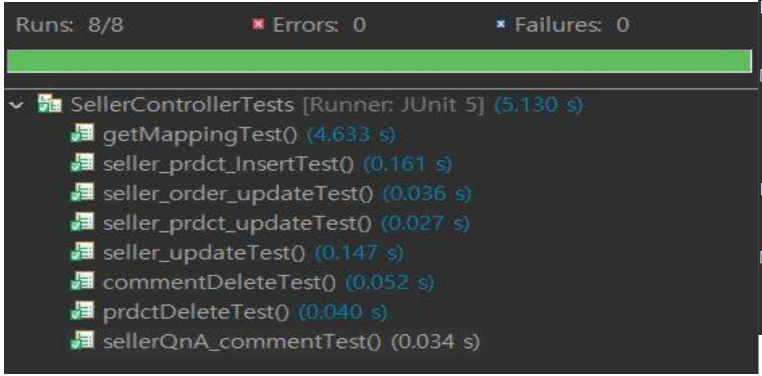
Runs: 8/8 Errors: 0 Failures: 0

```
SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```

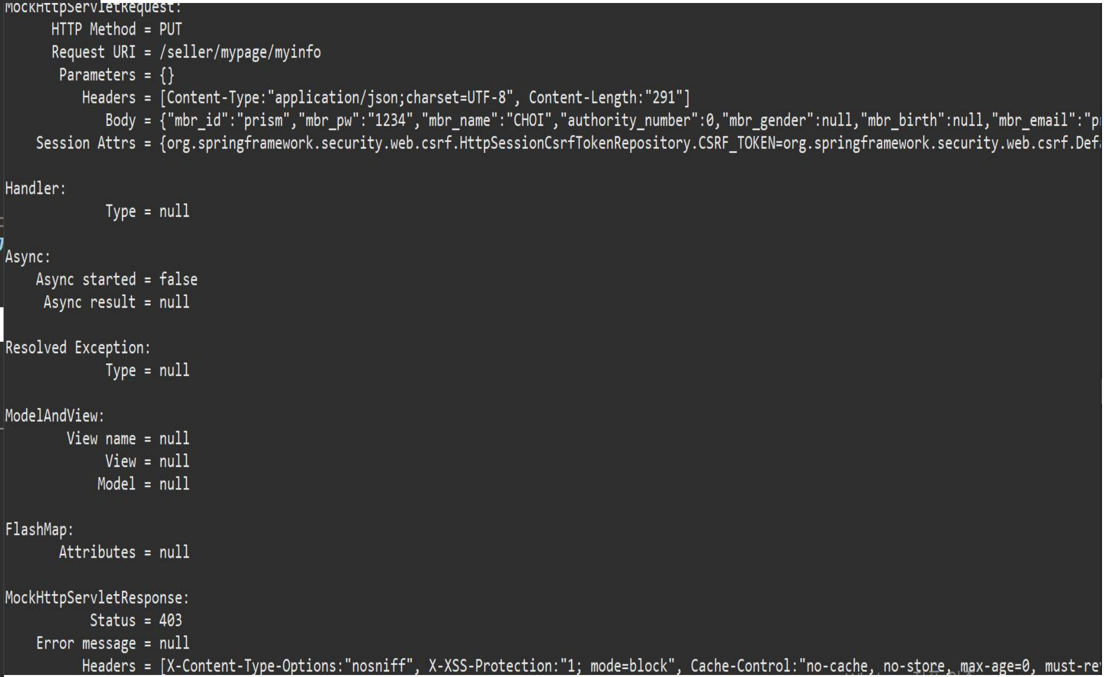
07 Seller 통합 테스트



```java
// 판매자 정보 수정 테스트
@Test
@WithUserDetails("prism")
public void seller_updateTest() throws Exception {
    MbrShippingVO msVO = new MbrShippingVO();
    msVO.setMbr_id("prism");
    msVO.setMbr_pw("1234");
    msVO.setMbr_name("CHOI");
    msVO.setMbr_email("prism@naver.com");
    msVO.setContact_number("01011111111");
    msVO.setShipping_address("서울시 종로구");

    String content = mapper.writeValueAsString(msVO);

    mockMvc.perform(put("/seller/mypage/myinfo") // postmapping t
            .content(content).contentType(MediaType.APPLICATION_J
            .andDo(print()).andReturn();
}
```



```
Runs:  8/8          ✖ Errors:  0          ✖ Failures:  0

⌄ SellerControllerTests [Runner: JUnit 5] (5.130 s)
    getMappingTest() (4.633 s)
    seller_prdct_InsertTest() (0.161 s)
    seller_order_updateTest() (0.036 s)
    seller_prdct_updateTest() (0.027 s)
    seller_updateTest() (0.147 s)
    commentDeleteTest() (0.052 s)
    prdctDeleteTest() (0.040 s)
    sellerQnA_commentTest() (0.034 s)
```



```
MockHttpServletRequest:
      HTTP Method = PUT
      Request URI = /seller/mypage/myinfo
       Parameters = {}
          Headers = [Content-Type:"application/json;charset=UTF-8", Content-Length:"291"]
             Body = {"mbr_id":"prism","mbr_pw":"1234","mbr_name":"CHOI","authority_number":0,"mbr_gender":null,"mbr_birth":null,"mbr_email":"p
    Session Attrs = {org.springframework.security.web.csrf.HttpSessionCsrfTokenRepository.CSRF_TOKEN=org.springframework.security.web.csrf.Def

Handler:
             Type = null

Async:
    Async started = false
     Async result = null

Resolved Exception:
             Type = null

ModelAndView:
        View name = null
             View = null
            Model = null

FlashMap:
       Attributes = null

MockHttpServletResponse:
           Status = 403
    Error message = null
          Headers = [X-Content-Type-Options:"nosniff", X-XSS-Protection:"1; mode=block", Cache-Control:"no-cache, no-store, max-age=0, must-re
```