

## 1

## Debug

## 오류 발견 - Access Error

The screenshot displays an IDE with two main panels. The left panel shows the source code of `CustomAccessDeniedHandler.java` and `security-context.xml`. The code is as follows:

```
1 package www.dream.com.party.security;
2
3 import java.io.IOException;
4
11
12 public class CustomAccessDeniedHandler implements AccessDeniedHandler {
13
14     @Override
15     public void handle(HttpServletRequest request, HttpServletResponse response,
16         AccessDeniedException accessDeniedException) throws IOException, ServletException {
17         accessDeniedException.printStackTrace();
18         response.sendRedirect("/accessError");
19     }
20 }
21 }
22
```

A breakpoint is set at line 17. The right panel shows the 'Variables' window with the following data:

Name	Value
no method return value	
this	CustomAccessDeniedHandler (id=110)
request	HttpServletRequestFactory\$Servlet3Securit...
response	HeaderWriterFilter\$HeaderWriterResponse (i...
accessDeniedException	AccessDeniedException (id=126)
backtrace	Object[6] (id=136)
cause	AccessDeniedException (id=126)
depth	60
detailMessage	"Access is denied" (id=138)
stackTrace	StackTraceElement[0] (id=144)
suppressedExceptions	Collections\$EmptyList<E> (id=145)

Below the variables table, there is a dropdown menu with the text '<Choose a previously entered expression>' and a text area containing the expression 'Access is denied'.

01

## AccessError #1

보통의 경우는 권한이 없어서 생기는 에러지만 만약 권한이 있음에도 Access Error가 발생할 경우, `AccessDeniedHandler`에 `accessDeniedException.printStackTrace();` 를 기입 후 breakpoint를 걸고 debug합니다.

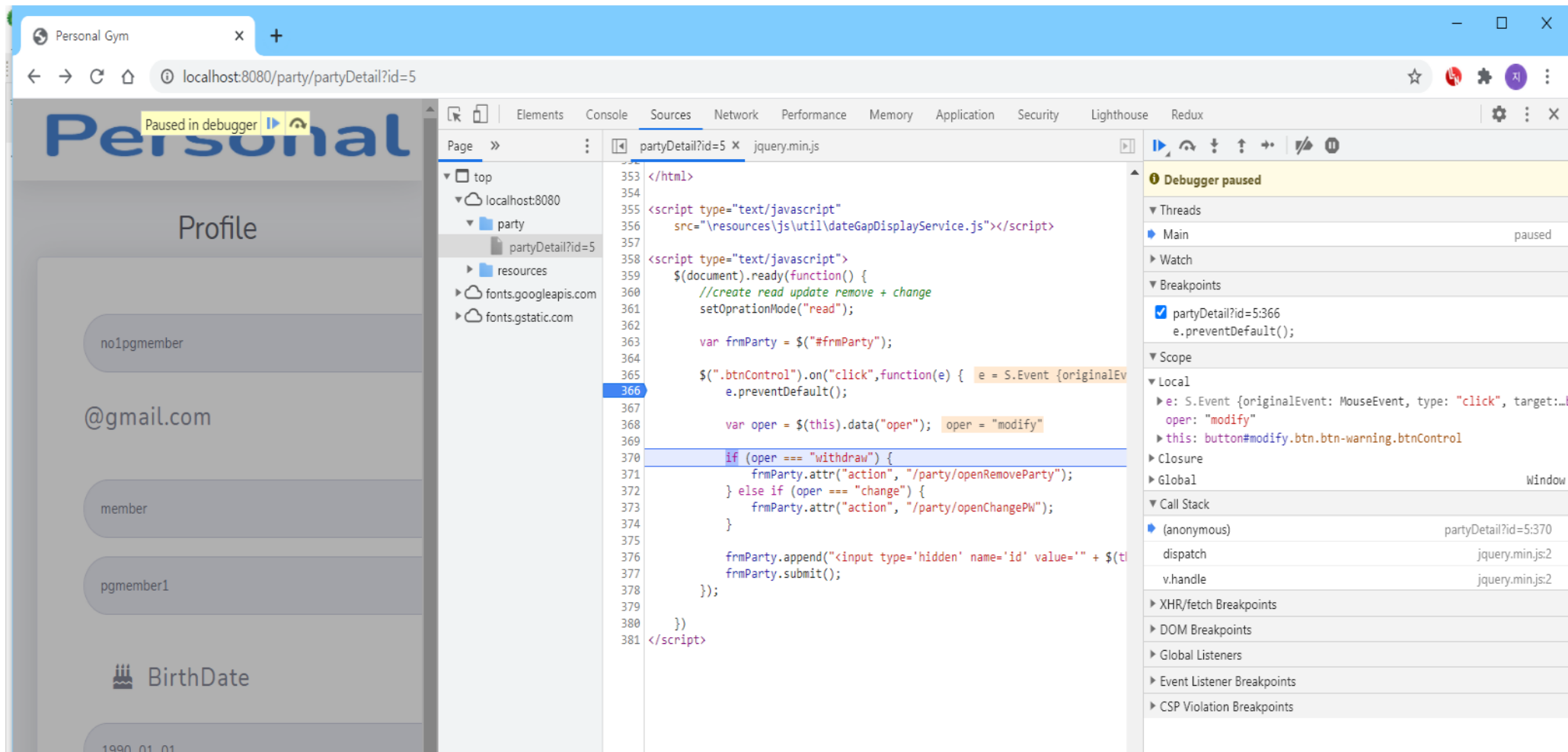
## 오류 발견 - Data 확인

```
193 @PostMapping("removeAdminParty")
194 public ResponseEntity<Integer> removeAdminParty(long[] arrTargetParty){
195     System.out.println(arrTargetParty);
196     return new ResponseEntity<>(partyService.removeAdminParty(arrTargetParty), HttpStatus.OK);
197 }
100
```

02 값의 출력 확인 #2  
controller에서 원하는 mapping에 breakpoint를 걸고 해당하는 변수/객체/배열에 마우스 커서를 올리거나 Variable 탭에서 확인 가능합니다.  
breakpoint의 위치가 혼동될 때 `system.out.println(mapping의 field명)`을 선언하여 sysout에 breakpoint걸고 debug합니다.

# 1 Debug

## 오류 발견 - Javascript Function 확인



### Javascript, Jsp #3

03

크롬 브라우저의 개발자도구(F12)에서 SOURCES 탭의 Function 내부에 있는 `e.preventDefault();`의 줄번호를 클릭하면, breakpoint를 걸 수 있습니다. 해당하는 function을 호출하여 확인 가능합니다.

localhost:8080 내용:

code:500

message:<!doctype html><html lang="ko"><head><title>HTTP  
상태 500 – 내부 서버 오류</title><style type="text/css">body  
{font-family:Tahoma,Arial,sans-serif;} h1, h2, h3, b  
{color:white;background-color:#525D76;} h1 {font-size:22px;} h2  
{font-size:16px;} h3 {font-size:14px;} p {font-size:12px;} a  
{color:black;} .line {height:1px;background-  
color:#525D76;border:none;}</style></head><body><h1>HTTP 상  
태 500 – 내부 서버 오류</h1><hr class="line" /><p><b>타입</  
b> 예외 보고</p><p><b>메시지</b> Request processing failed;  
nested exception is org.mybatis.spring.MyBatisSystemException:  
nested exception is org.apache.ibatis.builder.BuilderException: The  
expression &#39;array&#39; evaluated to a null value.</  
p><p><b>설명</b> 서버가, 해당 요청을 충족시키지 못하게 하는  
예기치 않은 조건을 맞닥뜨렸습니다.</p><p><b>예외</b></p></p>

```
org.springframework.web.util.NestedServletException:
Request processing failed; nested exception is
org.mybatis.spring.MyBatisSystemException: nested exception is
org.apache.ibatis.builder.BuilderException: The expression
'array' evaluated to a null value.
~org.springframework.web.servlet.FrameworkServlet.processReques
t(FrameworkServe
t.java:1014)
~org.springframework.web.servlet.FrameworkServlet.doPost(Framew
orkServlet.java:9
09)
~javax.servlet.http.HttpServlet.service(HttpServlet.java:652)
~org.springframework.web.servlet.FrameworkServlet.service(Framew
orkServlet.java:
883)
~javax.servlet.http.HttpServlet.service(HttpServlet.java:733)
...
```

확인

04

#### AJAX안의 오류 #4

```
error:function(request, status, error) {
    alert("code: " + request.status + "\n " + "message:" request.responseText + "\n " + "error: " + error);
}
```

이와 같은 `error:function` 코드를 추가하고 debug 하면 alert에서 오류를 확인 할 수 있습니다.

## 2

## JUnit Test

테스트 - findPerson

## Test Person

1. assertNotNull 이용해 partymapper가 null이 아닌지 확인합니다.

2. Foreach 문으로 모든 멤버(selectAllPerson)의 회원정보를 가져와 PersonVO의 person에 하나씩 넣고 출력합니다.

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src\\main\\webapp\\WEB-INF\\spring\\root-context.xml")
public class PartyMapperTest {
    @Autowired
    private PartyMapper partyMapper;

    @Test
    public void testPerson() {
        try {
            assertNotNull(partyMapper);
            for (PersonVO person : partyMapper.selectAllPerson())
                System.out.println(person);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Test
    public void testPersonWithCP() {
        try {
            assertNotNull(partyMapper);
            for (PersonVO person : partyMapper.selectAllPersonWithContactPoint())
                System.out.println(person);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems Javadoc Declaration Console Servers  
 <terminated> PartyMapperTest [JUnit] C:\Users\user-004\Downloads\sts-4.8.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_15.0.0.v20201014-1246\jre\bin\javaw.exe (2020. 12. 4. 5)

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".  
 SLF4J: Defaulting to no-operation (NOP) logger implementation  
 SLF4J: See <http://www.slf4j.org/codes.html#StaticLoggerBinder> for further details.  
 PersonVO [id=1, name=창업자, birthDate=Thu Jan 01 00:00:00 KST 1920, regDate=Fri Dec 04 1:  
 PersonVO [id=1, name=창업자, birthDate=Thu Jan 01 00:00:00 KST 1920, regDate=Fri Dec 04 1:



## 2

## JUnit Test

테스트 - findPost

## Test Post

findAllPost 함수 실행 시,

1) 1L은 게시글을 작성한 작성자 id(writer\_id)의 값이 1인 long타입을 나타냅니다.

2) 테스트 실행 시, id 값이 일치하는 writer가 작성한 게시글의 레코드를 모두 출력합니다.

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src\\main\\webapp\\WEB-INF\\spring\\root-context.xml")
public class ReplyMapperTest {
    @Autowired
    private ReplyMapper replyMapper;

    @Test
    public void testFindAllPost() {
        try {
            for (ReplyVO reply : replyMapper.findAllPost(1L))
                System.out.println(reply);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems Javadoc Declaration Console Servers  
 <terminated> ReplyMapperTest [JUnit] C:\Users\uer-004\Downloads\sts-4.8.1.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_15.0.0.v20201014-1246\jre\bin\javaw.exe (2020. 12. 8. 오전 10:23:07 - 오전 10:23:12)  
 PostVO [id=1, title=무궁한 발전 기원, content=많은 사람들에게 도움을..., writer=PersonVO [id=1, name=창업자, birthDate=null, regDa ^  
 PostVO [id=2, title=직원 모집 공고, content=열의에 가득찬 사람들 오세요, writer=PersonVO [id=1, name=창업자, birthDate=null, regDat

## 2

## JUnit Test

테스트 - findReply &amp; findReplyWithReply

```
@Test
public void testFindAllReply() {
    try {
        for (ReplyVO reply : replyMapper.findAllReply(2L))
            System.out.println(reply);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

SLF4J: Defaulting to no-operation (NOP) logger implementation  
 SLF4J: See <http://www.slf4j.org/codes.html#StaticLoggerBinder> for further details.

ReplyVO [id=3, content=findAllReply1Reply!, writer=PersonVO [id=1, name=창업자, birthDate=null, regDate=null, updateDate  
 ReplyVO [id=4, content=findAllReply2Reply!, writer=PersonVO [id=1, name=창업자, birthDate=null, regDate=null, updateDate

Test Reply

testFindAllReply 함수 실행 시,

1) 2L은 게시글의 id의 값이 2인 long타입을 나타냅니다.

2) 테스트 실행 시, id 값이 일치하는 게시글의 댓글 레코드를 출력합니다.

```
@Test
public void testFindReplyWithReply() {
    try {
        for (ReplyVO reply : replyMapper.findReplyWithReply(3L))
            System.out.println(reply);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

SLF4J: Defaulting to no-operation (NOP) logger implementation  
 SLF4J: See <http://www.slf4j.org/codes.html#StaticLoggerBinder> for further details.

ReplyVO [id=5, content=findReplyWithReply test 대댓글, writer=PersonVO [id=1, name=창업자, birthDate=null, regDate=null, updateDate

Test Re-Reply

testFindReplyWithReply 함수 실행 시,

1) 3L은 댓글의 id의 값이 3인 long타입을 나타냅니다.

2) 테스트 실행 시, id 값이 일치하는 댓글의 대댓글 레코드를 출력합니다.

## 2

## JUnit Test

테스트 - findHashTag

## Test HashTag

Foreach문을 사용하여 입력한 string 들이 Hashtag DB에 존재하는 지 확인합니다.

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src\\main\\webapp\\WEB-INF\\spring\\root-context.xml")
public class HasgTagMapperTest {
    @Autowired
    private HashTagMapper hashTagMapper;

    @Test
    public void findExisting() {
        try {
            for (HashTagVO hashTagVO : hashTagMapper.findExisting(new String[] {"IT", "자두", "영등포"}))
                System.out.println(hashTagVO);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Problems Javadoc Declaration Console Servers

<terminated> HasgTagMapperTest [JUnit] C:\Users\uer-004\Downloads\sts-4.8.1.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_15.0.0.v20201014-1246\jre\bin\javaw.exe (2020. 12. 14. 오후 5:48:18 - 오후 5:48:24)

SLF4J: See <http://www.slf4j.org/codes.html#StaticLoggerBinder> for further details.

HashTagVO(id=1, word=IT, descript=banana, superHashTag=null, childHashTag=null)