

게시판 구현하기 (예외처리)

게시판 구현하기 (상황-1)

게시판 구현하기 - 예외처리

• 상황-1

글쓰기 또는 글수정 할 때 제목이 오라클 MEMBER 컬럼의 길이보다 클 경우 에러 발생.
varChar2(50)보다 클 경우 예외 처리를 해주어야 함

HTTP 상태 500 – 내부 서버 오류

타입 예외 보고

메시지 Request processing failed; nested exception is org.apache.ibatis.exceptions.PersistenceException:

설명 서버가, 해당 요청을 충족시키지 못하게 하는 예기치 않은 조건을 맞닥뜨렸습니다.

예외

```
org.springframework.web.util.NestedServletException: Request processing failed; nested exception is org.apache.ibatis.exceptions.PersistenceException:
### Error updating database. Cause: java.sql.SQLException: ORA-12899: "SYSTEM"."BOARD"."B_TITLE" 열에 대한 값이 너무 큼(실제: 84, 최대값: 50)
```

```
### The error may involve boardMapper.insertBoard-Inline
### The error occurred while setting parameters
### SQL: INSERT INTO board(B_IDX, B_TITLE, B_CONTENT, B_WRITER, B_INDATE, B_COUNT) VALUES(seq_
### Cause: java.sql.SQLException: ORA-12899: "SYSTEM"."BOARD"."B_TITLE" 열에 대한 값이 너무 큼(실제: 84, 최대값: 50)

org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1014)
org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:909)
javax.servlet.http.HttpServlet.service(HttpServlet.java:660)
org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:883)
javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:201)
org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:117)
```

근본 원인 (root cause)

```
org.apache.ibatis.exceptions.PersistenceException:
### Error updating database. Cause: java.sql.SQLException: ORA-12899: "SYSTEM"."BOARD"."B_TITLE" 열에 대한 값이 너무 큼(실제: 84, 최대값: 50)
```

```
### The error may involve boardMapper.insertBoard-Inline
### The error occurred while setting parameters
### SQL: INSERT INTO board(B_IDX, B_TITLE, B_CONTENT, B_WRITER, B_INDATE, B_COUNT) VALUES(seq_
### Cause: java.sql.SQLException: ORA-12899: "SYSTEM"."BOARD"."B_TITLE" 열에 대한 값이 너무 큼(실제: 84, 최대값: 50)
```

게시판 구현하기 - 예외처리

• 참고

1. 이러한 예외 처리는 단순히 사용자에게 에러를 보여주지 않기 위한 목적도 있지만 DB 테이블 구조를 숨기기 위한 보안적인 측면도 있음
2. 클라이언트 측과 서버 측 모두 예외 처리를 해주는 것이 좋음
 - 클라이언트 측 : 사용자 경험 개선, 데이터 무효화 방지, 대부분의 예외 처리를 클라이언트에서 수행되기 때문에 불필요한 트래픽 및 서버 리소스 사용을 없앴
 - 서버 측 : 악의적인 사용자로부터의 변조 또는 위조된 데이터를 탐지 및 재확인

게시판 구현하기 - 예외처리

- `kr.co.greenart.common.controller.DataValidationController`

데이터 길이를 체크하는 로직 구현

```
@Controller
public class DataValidationController {

    public Boolean LanguageCheck(String data) {
        int byteLength = 0;

        for (char c : data.toCharArray()) {
            if (Character.toString(c).matches("[a-zA-Z]")) {
                byteLength += 1; // 영어는 1바이트로 계산
            } else if (Character.toString(c).matches("[ㄱ-ㅎ가-힣]")) {
                byteLength += 3; // 한글은 3바이트로 계산
            }
        }

        if(byteLength > 50) {
            return false;
        } else {
            return true;
        }
    }
}
```

게시판 구현하기 - 예외처리

- **kr.co.greenart.common.controller.DataValidationController**

1. 전달받은 문자열을 문자(char) 배열로 만들어 foreach를 통해 하나씩 문자를 꺼냄
2. 정규식으로 사용하기 위해 문자를 문자열로 만듦 (Character.toString)
3. 정규식을 통해 영어인지, 한글인지 확인
4. 영어는 1바이트, 한글은 3바이트로 계산하여 byteLength에 값을 더함
5. 50바이트가 넘어갈 경우 msg와 status를 세션에 저장

게시판 구현하기 - 예외처리

- **kr.co.greenart.board.controller.BoardController**

위에서 만든 컨트롤러에 데이터를 전달하고 true이면 정상적으로 실행하도록,
false이면 enrollForm.do 로 리다이렉션

```
@Autowired  
private DataValidationController dataValidation;
```

```
@PostMapping("insert.do")  
public String insertBoard(Board board, HttpSession session, Model model) {  
    board.setWriter((String) session.getAttribute("memberName"));  
  
    boolean titleLength = dataValidation.LanguageCheck(board.getTitle());  
  
    if(titleLength) {  
        int result = boardService.insertBoard(board);  
  
        if(result>0) {  
            session.setAttribute("msg", "등록 되었습니다.");  
            session.setAttribute("status", "success");  
            return "redirect:/board/list.do";  
        } else {  
            return "common/errorPage";  
        }  
    } else {  
        session.setAttribute("msg", "제목이 너무 길니다.");  
        session.setAttribute("status", "error");  
        return "redirect:/board/enrollForm.do";  
    }  
}
```

게시판 구현하기 - 예외처리

- **kr.co.greenart.board.controller.BoardController**

enrollForm.do는 요청이 올 경우 전달받은 세션이 있다면 팝업창 출력

```
@GetMapping("enrollForm.do")
public String enrollForm(Model model, HttpSession session) {
    model.addAttribute("msg", (String) session.getAttribute("msg"));
    model.addAttribute("status", (String) session.getAttribute("status"));

    session.removeAttribute("msg");
    session.removeAttribute("status");

    return "board/boardEnroll";
}
```


게시판 구현하기 - 예외처리

- 본문 내용에 대한 예외 처리 작성 해보기

테스트를 위해 100바이트를 기준으로 본문 내용이 넘어갈 경우 팝업 창 출력하기

게시판 구현하기 - 예외처리

• 참고

1. CLOB 타입은 일반적으로 최대 4GB까지 사용이 가능하나, 현실적으로 4GB를 매번 검사하기에는 성능 이슈가 발생할 가능성이 높음
2. 사용자가 입력 가능한 최대 데이터의 크기를 예측하여 정해놓고 설정
ex) 사용자는 사진 등을 업로드 해도 최대 500MB를 넘기지 않을 것이라 생각되어
500MB → 524288000로 환산하여 연산 수행 등

게시판 구현하기 (상황-2)

게시판 구현하기 - 예외처리

• 상황-2

글 제목 또는 내용을 입력하지 않았을 경우에 대한 예외 처리가 필요함

HTTP 상태 500 – 내부 서버 오류

타입 예외 보고

메시지 Request processing failed; nested exception is org.springframework.dao.DataIntegrityViolationException:

설명 서버가, 해당 요청을 충족시키지 못하게 하는 예기치 않은 조건을 맞닥뜨렸습니다.

예외

```
org.springframework.web.util.NestedServletException: Request processing failed; nested exception is org.springframework.dao.DataIntegrityViolationException:
### Error updating database. Cause: java.sql.SQLIntegrityConstraintViolationException: ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다

### The error may involve boardMapper.insertBoard-Inline
### The error occurred while setting parameters
### SQL: INSERT INTO board(B_ID, B_TITLE, B_CONTENT, B_WRITER, B_INDATE, B_COUNT) VALUES(seq_b_idx.I
### Cause: java.sql.SQLIntegrityConstraintViolationException: ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다

; ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다
; nested exception is java.sql.SQLIntegrityConstraintViolationException: ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다

org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1014)
org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:909)
javax.servlet.http.HttpServlet.service(HttpServlet.java:660)
org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:883)
javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:201)
org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:117)

근본 원인 (root cause)

org.springframework.dao.DataIntegrityViolationException:
### Error updating database. Cause: java.sql.SQLIntegrityConstraintViolationException: ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다

### The error may involve boardMapper.insertBoard-Inline
### The error occurred while setting parameters
### SQL: INSERT INTO board(B_ID, B_TITLE, B_CONTENT, B_WRITER, B_INDATE, B_COUNT) VALUES(seq_b_idx.I
### Cause: java.sql.SQLIntegrityConstraintViolationException: ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다

; ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다
; nested exception is java.sql.SQLIntegrityConstraintViolationException: ORA-01400: NULL을 ("SYSTEM"."BOARD"."B_TITLE") 안에 삽입할 수 없습니다
```

게시판 구현하기 - 예외처리

- **kr.co.greenart.common.controller.DataValidationController**

전달받은 값이 비어 있는지 확인하는 로직 구현

```
public Boolean nullCheck(String data) {  
    if(data.isEmpty()) {  
        return false;  
    } else {  
        return true;  
    }  
}
```

게시판 구현하기 - 예외처리

- kr.co.greenart.board.controller.BoardController

값이 비어 있는지 확인 후 해당하는 상황에 따라 처리하는 조건식

```
@PostMapping("insert.do")
public String insertBoard(Board board, HttpSession session, Model model) {
    board.setWriter((String) session.getAttribute("memberName"));

    boolean titleLength = dataValidation.LanguageCheck(board.getTitle());
    boolean titleNull = dataValidation.nullCheck(board.getTitle());

    if(titleLength && titleNull) {
        int result = boardService.insertBoard(board);

        if(result>0) {
            session.setAttribute("msg", "등록 되었습니다.");
            session.setAttribute("status", "success");
            return "redirect:/board/list.do";
        } else {
            return "common/errorPage";
        }
    } else if(!titleLength){
        session.setAttribute("msg", "제목이 너무 길니다.");
        session.setAttribute("status", "error");
        return "redirect:/board/enrollForm.do";
    } else if(!titleNull) {
        session.setAttribute("msg", "제목을 입력해주세요.");
        session.setAttribute("status", "error");
        return "redirect:/board/enrollForm.do";
    } else {
        session.setAttribute("msg", "에러가 발생했습니다.");
        session.setAttribute("status", "error");
        return "redirect:/board/enrollForm.do";
    }
}
```

게시판 구현하기 - 예외처리

- 본문 내용에 대한 예외 처리 작성 해보기

본문 내용이 비어있을 경우 처리하는 코드 작성 해보기

게시판 구현하기 (중복 코드 제거)

게시판 구현하기 - 중복 코드 제거

• 중복 코드 제거

여러가지 상황으로 인해 발생하는 중복 코드를 줄이기 위한 메서드를 생성하고 중복 코드를 제거 해보세요.

패키지 명 : kr.co.greenart.common.controller.SessionController

메서드 명 : setSessionMessage

```
@PostMapping("insert.do")
public String insertBoard(Board board, HttpSession session, Model model) {
    board.setWriter((String) session.getAttribute("memberName"));

    boolean titleLength = dataValidation.LanguageCheck(board.getTitle());

    boolean titleNull = dataValidation.nullCheck(board.getTitle());

    if(titleLength && titleNull) {
        int result = boardService.insertBoard(board);

        if(result>0) {
            session.setAttribute("msg", "등록 되었습니다.");
            session.setAttribute("status", "success");
            return "redirect:/board/list.do";
        } else {
            return "common/errorPage";
        }
    } else if(!titleLength){
        session.setAttribute("msg", "제목이 너무 길니다.");
        session.setAttribute("status", "error");
        return "redirect:/board/enrollForm.do";
    } else if(!titleNull) {
        session.setAttribute("msg", "제목을 입력해주세요.");
        session.setAttribute("status", "error");
        return "redirect:/board/enrollForm.do";
    } else {
        session.setAttribute("msg", "에러가 발생했습니다.");
        session.setAttribute("status", "error");
        return "redirect:/board/enrollForm.do";
    }
}
```