2021.06.17 mySQL + HTML/JS/CSS

gradle (demo) × application.yaml ×

```yaml
server:
  port: 7777

spring:
  datasource:
    url: jdbc:mysql://localhost:3306/non_jpa_db?serverTimezone=UTC@useSSL=false
    username: khweb
    password: kh[redacted]
    dirver-class-name: com.mysql.cj.jdbc.Driver
```

mySQL 가동한 후에

yaml 파일에 datasource 관련 추가.

<< 오타 수정 해야됨

UTC&useSSL=false

@이 아니고 &

두산백과

JDBC

요약 자바 프로그램 안에서 SQL을 실행하기 위해 데이터베이스를 연결해주는 응용프로그램 인터페이스를 말한다.

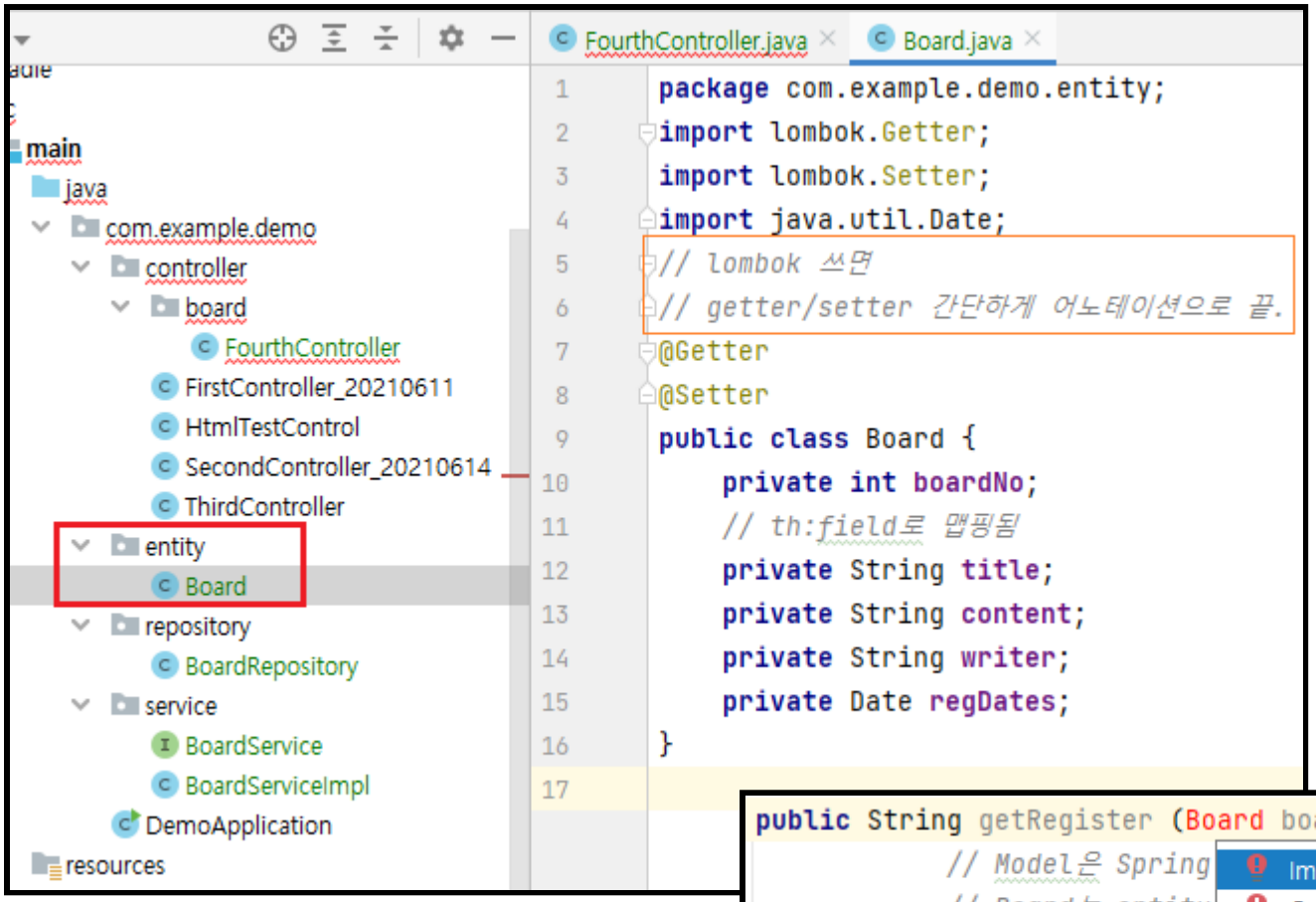
자바 프로그램 내에서 데이터베이스 질의문 즉, SQL을 실행하기 위한 자바 API(application programming interface)이다. Java database connectivity의 약자로 생각하기도 하지만 실제로는 상표 이름이다. JDBC는 데이터베이스 및 애플리케이션 개발자들을 위한 표준 API를 제공하고 순수 자바 API만으로도 데이터베이스 응용업무를 만들게 해준다.

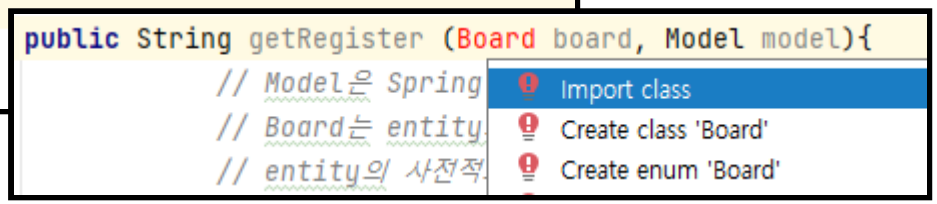
```java
package com.example.demo.controller.board;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
@Slf4j
@Controller
public class FourthController {
    // @Autowired > 스프링이 자동으로 객체를 찾을 수 있게 support
    @Autowired
    private BoardService service;
    // thymeleaf에서는 아래와 같이 특정한 객체를 입력으로 받으면
    // HTML에서 th:object와 같은 키워드를 통해 정보를 획득할 수 있다.
    // 즉 board 객체의 정보를 획득할 수 있음을 의미.
    @GetMapping("/register")
    public String getRegister (Board board, Model model){
            // Model은 Spring 내부에 있어서 상관 없는데
            // Board는 entity로 따로 만들어줘야 한다.
            // entity의 사전적의미 : 독립체
        log.info("getRegister()");
        return "/board/Fourth_20210617/register";
    }

    @PostMapping("/register")
    public String postRegister (Board board, Model model){
        log.info("postRegister()");
        service.register(board);
        model.addAttribute( attributeName: "msg", attributeValue: "등록이 완료되었습니다");
                        // 'msg' 전달을 위한 코드
        return "/board/Fourth_20210617/success";
    }
```

```
mysql> show grants for 'khweb'@'localhost';
+---------------------------------------------------------------+
| Grants for khweb@localhost                                    |
+---------------------------------------------------------------+
| GRANT USAGE ON *.* TO `khweb`@`localhost`                     |
| GRANT ALL PRIVILEGES ON `non_jpa_db`.* TO `khweb`@`localhost` |
+---------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql> show grants for 'khweb@localhost';
ERROR 1141 (42000): There is no such grant defined for user 'khweb@localhost' on host '%'
mysql> show grants for 'khweb@localhost';
ERROR 1141 (42000): There is no such grant defined for user 'khweb@localhost' on host '%'
mysql> use non_jpa_db;
Database changed
mysql> create table board(
    ->     board_no int not null auto_increment,
    ->     title varchar(200) not null,
    ->     content text null,
    ->     writer varchar(50) not null,
    ->     reg_date timestamp not null default now(),
    ->     primary key(board_no)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql>
```

우리는 mySQL이용해서 db에 만든 'board'의 정보와 Spring을
연결시키려고 하는중
따라서 board_no / title / content /... ... / reg_date의
type들에 맞춰서 작성해야됨.

다음 page entity로 board 만든 것 확인





<< 이 data들을 사용하려면
getter와 setter가 필요함.

entity로 board class 만들어 줬으니
controller 쪽에 import

이제 html 파일로 register / success page를 만들 것

java × register.html ×

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<!-- 버튼 누르면 동작하는 과정을 서포트하는 라이브러리 -->
<script type="text/javascript" src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script> // 이쪽 부분은 RUN-TIME에 구동되는 부분임 주의
             >> 인텔리J TEST로 구동시키면 이 부분은 체크 안 함

   // document는 현재 웹 상에 떠 있는 page 자체를 의미한다고 보면 됨.
     $(document).ready(function(){

       // form 태그에 id="board"에 해당하는 객체를 얻어옴.
       // body 안에 있는 id가 board인 <form> 태그 자체를 객체화 시켜서
       // formObj라는 변수에 대입한다는 의미
         var formObj = $("#board");

       // id="btnRegister"를 클릭했을 때
       $("#btnRegister").on("click", function(){
           // URL "register"로 보낸다, method는 "post"
           formObj.attr("action", "register");
           formObj.attr("method", "post");
           formObj.submit();                    #btnList에 연동시킬 것은
       });                                      다음시간에 작성할듯
     })
  </script>
<body>
```

《 source 찾은 방법은 아래에 따로 게시

<script> 내부에 버튼을 누르면 동작하도록 하는
로직들이 있는데
이 것들이 실행 가능한 이유가
source로 따온 jqeury 덕분이다.

Q. 위 설명이 맞나요??

// Post방식은 data를 어떤 식으로 관리하는가 》 json
// json 형식은 data형식이
hashmap과 비슷한 느낌으로 전송됨

// ***그렇다면 PostMappin을 쓰면 좋은점?***

GetMapping으로 하면 id/비밀번호 같은게 url에 다 나옴
PostMapping은 json의 data형식으로 나가기 때문에 보호됨.

《 script쪽 마지막에 오타 수정해야됨. - ; 빠졌음
그리고 위에 인텔리J TEST 앞에 주석처리 // 해줘야됨

```html
<body>
  <h2> 게시글 등록</h2>
  <!--controller에 주석으로 thymeleaf에 관하여 기재하였듯이
  아래의 {board}는 Controller의 getRegister에서 전달받은 Board 객체(현재 텅 빔)
   th:action에는 이동시킬 URL 주소를 작성.
   보통 action에는 @를 사용
   th: object는 Board(텅 빔)객체를 전달받기 위한 목적으로 활용
   객체를 전달 받을 경우엔 $를 사용.-->
  <form id="board" th:action="@{register}" th:object="${board}" method="post">
    <!-- 텅 빈 객체 안에 data들을 넣는 작업 시작 -->
    <table>
      <tr>
        <td>제목</td>
        <!-- th:field의 경우엔 Board 객체에 있는 title과 직접 맵핑시킴 -->
        <td><input type="text" name="title" th:field="*{title}"></td>
      </tr>
      <tr>
        <td>작성자</td>
        <!-- th:field의 경우엔 Board 객체에 있는 writer와 직접 맵핑시킴 -->
        <td><input type="text" name="writer" th:field="*{wrtier}"></td>
      </tr>
      <tr>
        <td>본문</td>
        <!-- textarea는 글자를 여러개 입력할 수 있는 글 입력창이다 -->
        <td><textarea cols="50" rows="20" name="content" th:field="*{content}"></textarea></td>
      </tr>
    </table>
  </form>
  <div>
    <button type="submit" id="btnRegster"> 게시글 등록 </button>
    <button type="submit" id="btnList"> 게시글 목록 보기 </button>
  </div>
</body>
```

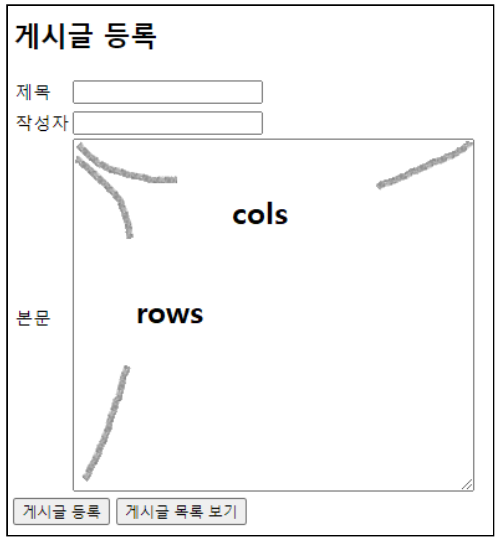
Q. 이미 SCRIPT 쪽에서 PostMapping된 register로 이동되는 로직을 만들어 놨는데 〈form〉 태그 내부에도 thymeleaf를 사용해서 갖은 로직을 넣어 놓는 이유가 뭔가요??

$: data 전달 받을 때(단방향)

@: 이동

***: data의 양방향 상호작용**

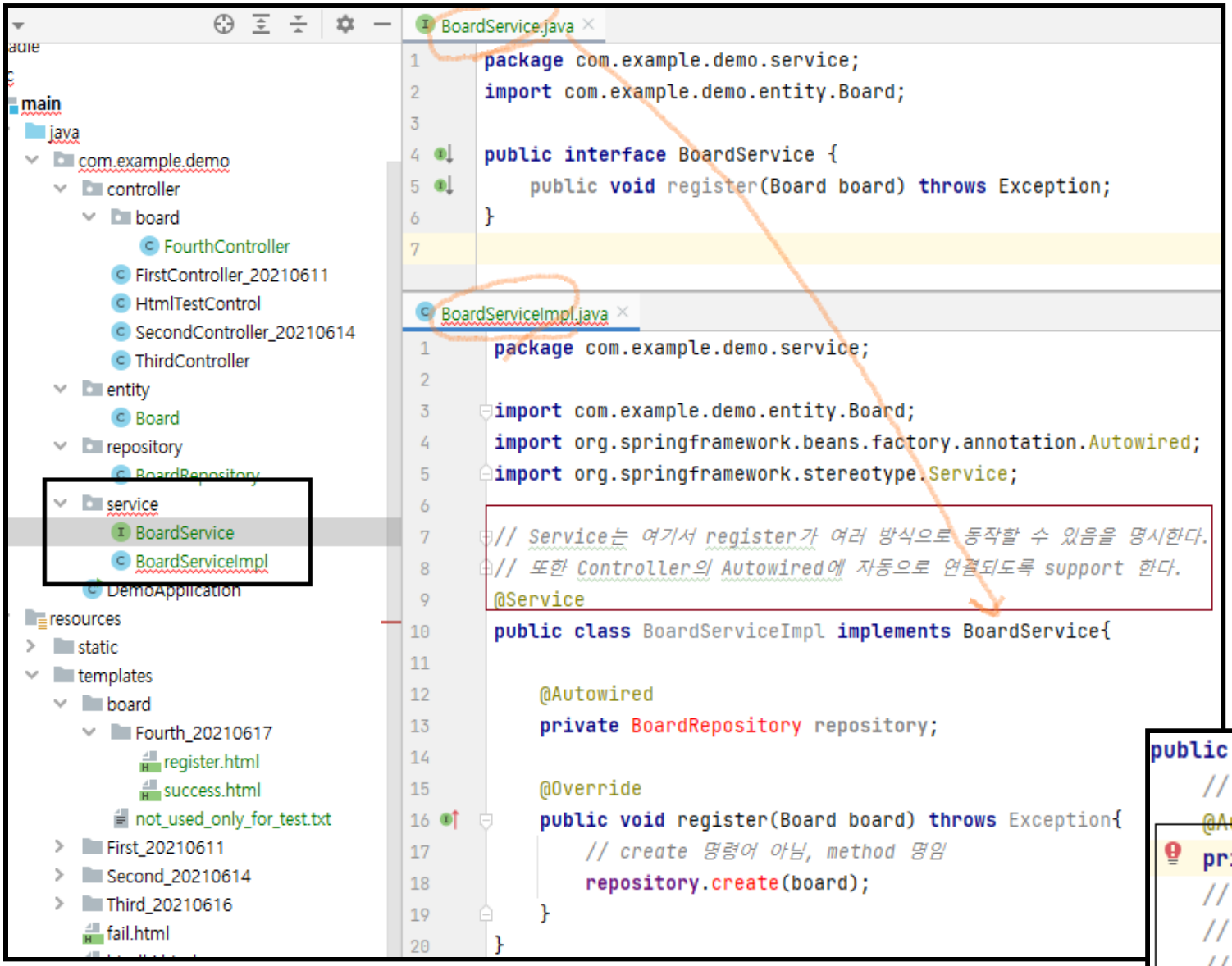
〈〈 btnRegister로 오타 수정



d.java × register.html × success.html ×

```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
  <h2>동작 성공</h2>
  <div>
      <!--
        MVC(Model View Contorller) Pattern
        Model: 다루는 data
        View: 눈에 보이는 화면
        Controller: URL 제어
      -->
      <!--${msg}를 통해 전달받은 Model의 msg 속성에 등록된 값을 활용하는 부분 -->
    <h3 th:text="${msg}"></h3>
  </div>
</body>
</html>
```

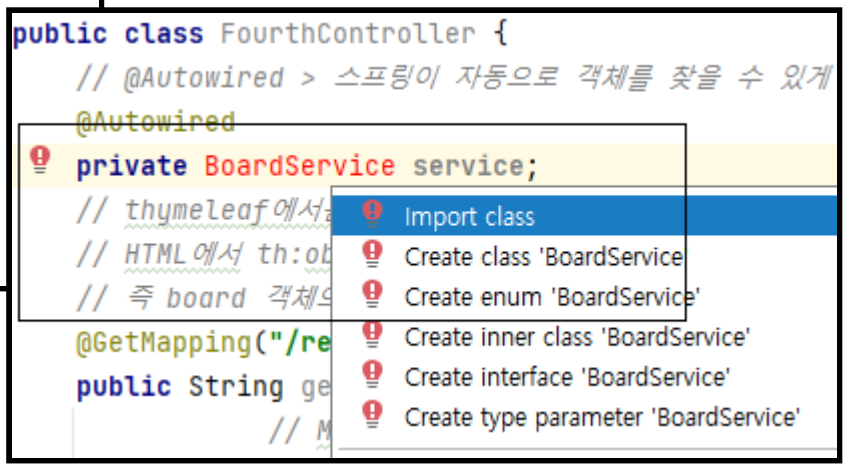
```java
@PostMapping("/register")
public String postRegister (Board board, Model model){
    log.info("postRegister()");
    //service.register(board);
    model.addAttribute( attributeName: "msg", attributeValue: "등록이 완료되었습니다");
        // 'msg' 전달을 위한 코드
        // key: msg, value: "등록이 완료되었습니다" 라고 생각하면 편할듯
    return "/board/Fourth_20210617/success";
}
```



repository.create(board);

사용자의 입력으로 넘어온 board 정보를

db에 저장하기 위해 db를 생성하는 작업



```java
package com.example.demo.repository;
import com.example.demo.entity.Board;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Repository;
@Repository
public class BoardRepository {
    @Autowired
    private JdbcTemplate jdbcTemplate;
    public void create(Board board) throws Exception {
        //db 처리를 하기 때문에 exception 넣어줘야됨

        //insert into board: DB에 있는 BOARD 테이블에 값을 집어넣겠다.
        //(title, content, writer): board 테이블 내에 있는 컬럼들
        // values (?, ?, ?): 뭔가 값을 넣을 것인데 아직 미정이라는 의미
        String query = "insert into board (title, content, writer) values (?, ?, ?)";

        //jdbcTemplate.update(): 이것을 통해 실제 DB상의 값을 갱신
        //query: 구동시킬 DB의 쿼리
        jdbcTemplate.update(query, board.getTitle(), board.getContent(), board.getContent());
        // board객체안에 사용자의 입력으로 받은 title/content/content가 있고
        // 그걸 get해서 ? ? ? 에 넣겠다는 의미
```

((getWriter로 오타수정

마지막으로 controller 쪽에 service 부분 살려줌

```java
    @PostMapping("/register")
    public String postRegister (Board board, Model model){
        log.info("postRegister()");
        service.register(board);
        model.addAttr
            // 'msg'
            // key: m
        return "/boar
    }
}
// Post방식은 data를 어
// json 형식은 data형식
//  그렇다면 PostMappi
```

Unhandled exception: java.lang.Exception

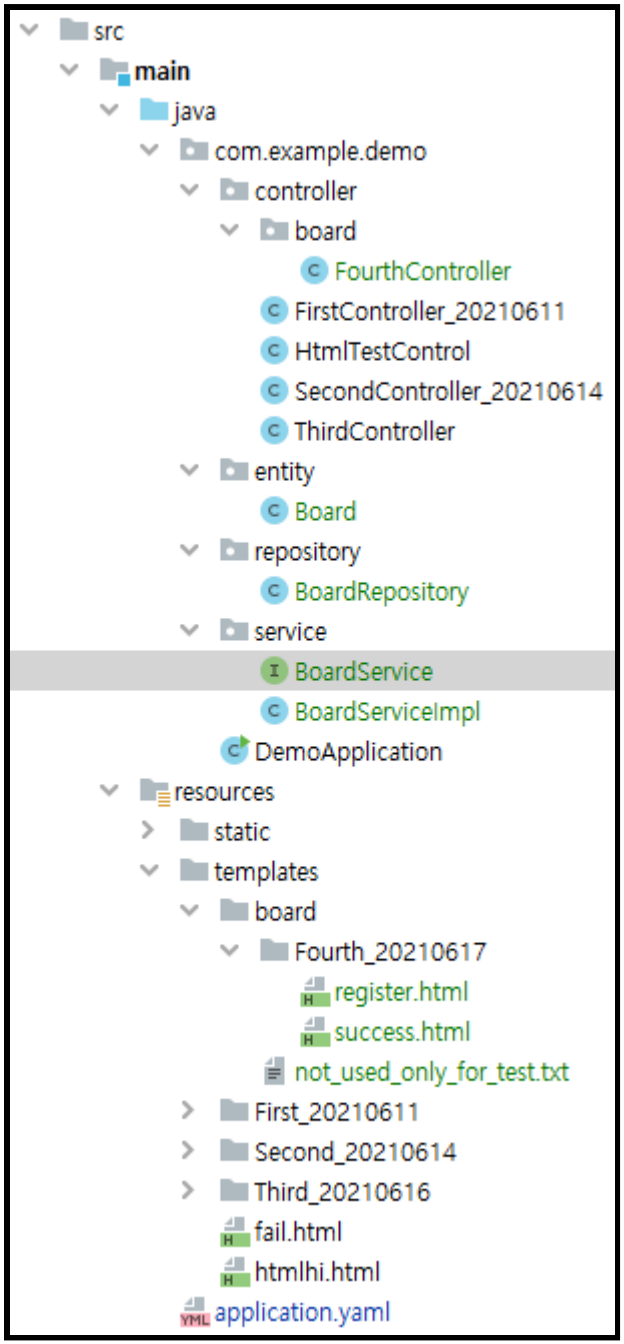
Add exception to method signature Alt+Shift+Enter Mor

com.example.demo.service.BoardService
public void **register**(Board board)
throws Exception

Throws: Exception

· demo.main

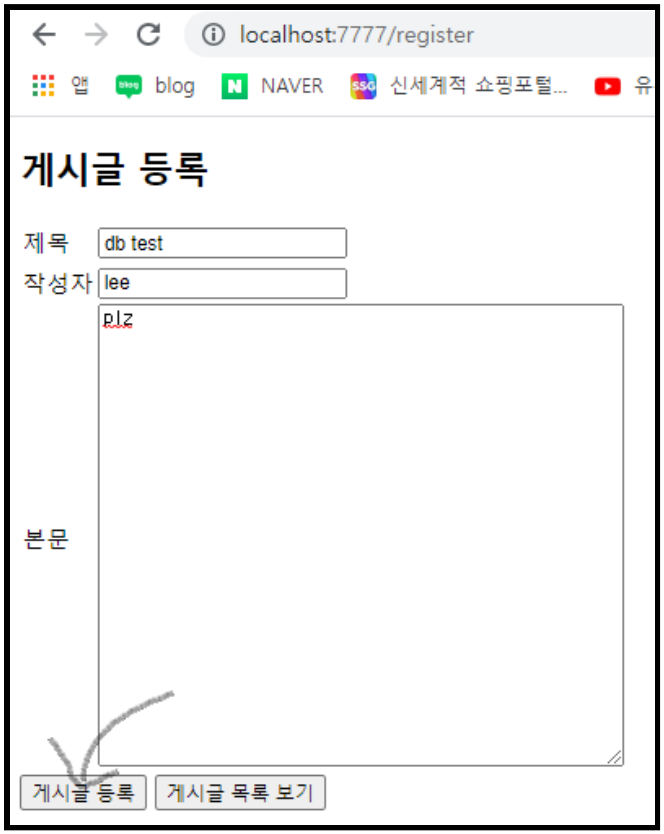
```java
@PostMapping("/register")
public String postRegister (Board board, Model model) throws Exception {
    log.info("postRegister()");
    service.register(board);
    model.addAttribute( attributeName: "msg", attributeValue: "등록이 완료되었습니다");
        // 'msg' 전달을 위한 코드
        // key: msg, value: "등록이 완료되었습니다" 라고 생각하면 편할듯
    return "/board/Fourth_20210617/success";
}
```


src
main
java
com.example.demo
controller
board
FourthController
FirstController_20210611
HtmlTestControl
SecondController_20210614
ThirdController
entity
Board
repository
BoardRepository
service
BoardService
BoardServiceImpl
DemoApplication
resources
static
templates
board
Fourth_20210617
register.html
success.html
not_used_only_for_test.txt
First_20210611
Second_20210614
Third_20210616
fail.html
htmlhi.html
application.yaml

저장 경로들 확인.

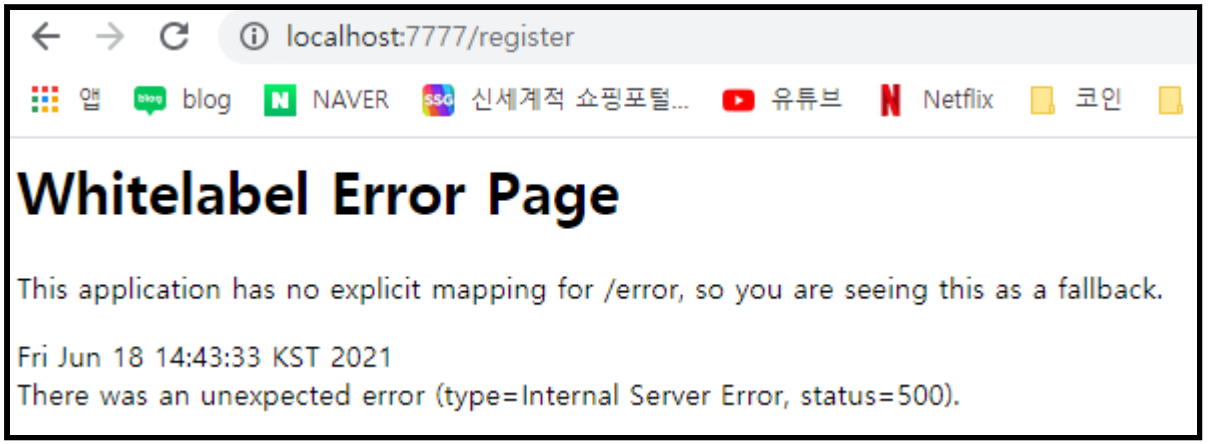
Q. ‹‹강사님께 문의하고 업데이트 할 부분›› 수업 중에는 이해가 잘 안 가서 따로 코드 적지 않고 강사님 수업 내용 따라가려고 했고 수업 끝나고 혼자서 코드 작성해서 확인해보니 다음과 같은 error 나타납니다. ››› 해결됨. 제발 오타 잘 확인 할 것 두시간 걸림

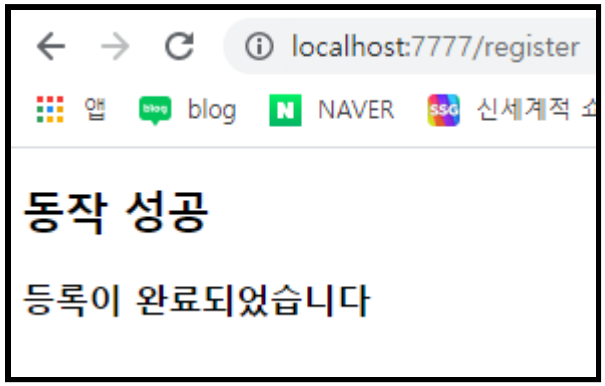
yaml 파일에도 오타나있었음. run 부분에 error 정의 잘 보면서 error 핸들링 할 것



게시글 등록을 누르면

아래와 같은 error가 발생합니다.





success.html 나옴 (주소는 PostMapping이기 때문에 7777/register로 나옴)

```
mysql> select * from board;
+----------+---------+---------+--------+---------------------+
| board_no | title   | content | writer | reg_date            |
+----------+---------+---------+--------+---------------------+
|        1 | db test | plz     | plz    | 2021-06-18 15:03:18 |
+----------+---------+---------+--------+---------------------+
1 row in set (0.00 sec)

mysql> select * from board;
+----------+---------+---------+--------+---------------------+
| board_no | title   | content | writer | reg_date            |
+----------+---------+---------+--------+---------------------+
|        1 | db test | plz     | plz    | 2021-06-18 15:03:18 |
|        2 | db test | whywhy  | whywhy | 2021-06-18 15:06:49 |
+----------+---------+---------+--------+---------------------+
2 rows in set (0.00 sec)

mysql> select * from board;
+----------+---------+---------+--------+---------------------+
| board_no | title   | content | writer | reg_date            |
+----------+---------+---------+--------+---------------------+
|        1 | db test | plz     | plz    | 2021-06-18 15:03:18 |
|        2 | db test | whywhy  | whywhy | 2021-06-18 15:06:49 |
|        3 | real    | plz     | lee    | 2021-06-18 15:10:22 |
+----------+---------+---------+--------+---------------------+
3 rows in set (0.00 sec)
```

db에 계속 writer에도 content 내용 입력되서 오타 찾아내고 해결 함.

```
public class BoardRepository {
    @Autowired
    private JdbcTemplate jdbcTemplate;
    public void create(Board board) throws Exception {
        //db 처리를 하기 때문에 exception 넣어줘야됨

        //insert into board: DB에 있는 BOARD 테이블에 값을 집어넣겠다.
        //(title, content, writer): board 테이블 내에 있는 컬럼들
        // values (?, ?, ?): 뭔가 값을 넣을 것인데 아직 미정이라는 의미
        String query = "insert into board (title, content, writer) values (?, ?, ?)";

        //jdbcTemplate.update(): 이것을 통해 실제 DB상의 값을 갱신
        //query: 구동시킬 DB의 쿼리
        jdbcTemplate.update(query, board.getTitle(), board.getContent(), board.getContent());
        // board객체안에 사용자의 입력으로 받은 title/content/content가 있고
        // 그걸 get해서 ? ? ? 에 넣겠다는 의미
```

```
Type 'help;' or '₩h' for help. Type '₩c' to clear the current input statement.

mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> use non_jpa_db;
Database changed
mysql> show tables;
+----------------------+
| Tables_in_non_jpa_db |
+----------------------+
| board                |
+----------------------+
1 row in set (0.02 sec)

mysql> desc board;
+----------+--------------+------+-----+-------------------+-------------------+
| Field    | Type         | Null | Key | Default           | Extra             |
+----------+--------------+------+-----+-------------------+-------------------+
| board_no | int          | NO   | PRI | NULL              | auto_increment    |
| title    | varchar(200) | NO   |     | NULL              |                   |
| content  | text         | YES  |     | NULL              |                   |
| writer   | varchar(50)  | NO   |     | NULL              |                   |
| reg_date | timestamp    | NO   |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+----------+--------------+------+-----+-------------------+-------------------+
5 rows in set (0.01 sec)

mysql> select * from board
    -> select * from board;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that co
 for the right syntax to use near 'select * from board' at line 2
mysql> select * from board;
Empty set (0.01 sec)

mysql> select * from board;
+----------+---------+---------+--------+---------------------+
| board_no | title   | content | writer | reg_date            |
+----------+---------+---------+--------+---------------------+
|        1 | db test | plz     | plz    | 2021-06-18 15:03:18 |
+----------+---------+---------+--------+---------------------+
1 row in set (0.00 sec)
```

<< 사용할 db 선택 먼저 해야됨.

저거 안 하고 계속 show table 하는데 안 돼서 놀랬음.