

Spring 월말

1. pom 복붙 > root test로 바꾸기

```
<properties>
  <java-version>1.8</java-version>
  <m2eclipse.wtp.contextRoot>test</m2eclipse.wtp.contextRoot>
  <!-- maven compiler plugin version :: https://maven.apache.
```

2. web.xml

```
리스터 밑에
<!-- POST 방식의 한글 처리. -->
  <filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

<!-- Processes application requests -->
<servlet>
  <servlet-name>appServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
  </init-param>
  <!-- DispatcherServlet이 해당 mapping을 찾지 못할 경우 NoHandlerFoundException를 throw하게 설정 -->
  <init-param>
    <param-name>throwExceptionIfNoHandlerFound</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>appServlet</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>

</web-app>
```

3. root-context (board > test로 변경)

```
<context:component-scan
    base-package="com.ssafy.*.model,com.ssafy.util,com.ssafy.board.aop" />
```

4. Meta-INF or root-context DB 이름 변경

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
    <Resource name="jdbc/ssafy" auth="Container" type="javax.sql.DataSource"
        maxTotal="100" maxIdle="30" maxWaitMillis="10000"
        username="ssafy" password="ssafy" driverClassName="com.mysql.cj.jdbc.
        url="jdbc:mysql://localhost:3306/ssafyweb?serverTimezone=UTC&useU
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>
```

```
<property name="url"
    value="jdbc:mysql://127.0.0.1:3306/ssafyweb?serverTimezone=UTC&useU
```

```
<property name="driverClass" value="com.mysql.cj.jdbc.Driver" />
<property name="url"
    value="jdbc:mysql://127.0.0.1:3306/ssafysample?serverTimezon
```

5. 리소시스/매퍼 통째로 복붙 후 매퍼안에 다 지우고 경로 변경

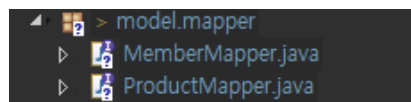
```
4
5 <mapper namespace="com.ssafy.test.model.mapper.ProductMapper">
6
7 </mapper>
```

6. jsp 파일 탐색기에서 복붙 후 /WEB-INF/views 경로 수정

```
<%@ include file="/WEB-INF/views/include/head.jsp" %>
```

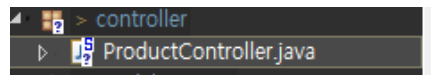
7. 리소시스/매퍼에서 경로 복사해서 Mapper 인터페이스 생성

- a. 만들자마자 @Mapper



8. 컨트롤러 생성

- a. 만들자마자 @Controller



9. Dto 변수들 칼럼명이랑 일치하게 만들기

```
private String code;
private String name;
private int price;
private String id;
```

10. nav.jsp a태그 경로 변경

a. `${root}/regist` or `/test/regist` 알아서 판단해서 바꾸기

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <ul class="navbar-nav me-auto">
    <li class="nav-item"><a class="nav-link" href="#">상품 목록</a></li>
    <li class="nav-item"><a class="nav-link" href="${root}/regist">등록</a></li>
  </ul>
</nav>
```

11. regist.jsp from name 넣기(Dto 변수명이랑 같게)

```
<label for="productCode">고유번호</label>
<input type="text" class="form-control" name="code" id="productCode" placeholder="고유번호" />
</div>
<div class="form-group">
  <label for="model">모델명</label>
  <input type="text" class="form-control" name="model" id="model" placeholder="모델명" />
</div>
<div class="form-group">
  <label for="price">가격</label>
  <input type="number" class="form-control" name="price" id="price" placeholder="가격" />
</div>
<div class="form-group">
  <label for="userId">등록자명</label>
  <input type="text" class="form-control" name="id" id="userId" placeholder="등록자명" />
</div>
```

12. input hidden 지우고 from에 post action바꾸기

```
<form id="registForm" method="post" action="${root }/regist">
  <div class="form-group">
```

13. 상품등록

```
<input type="hidden" name="action" value="${root}/regist">
```

```
@PostMapping("/regist")
public String regist(ProductDto productDto) {
    System.out.println(productDto.toString());
    return "regist";
}
```

14. 예외 처리는 throws Exception

15. 서비스임플 매퍼 선언

```
private final ProductMapper productMapper;

public ProductServiceImpl(ProductMapper productMapper) {
    super();
    this.productMapper = productMapper;
}

@Override
public void regist(ProductDto productDto) throws Exception {
    // TODO Auto-generated method stub
    productMapper.regist(productDto);
}
```

16. 매퍼 인터페이스에 메서드 만들었으면 product.xml에 쿼리문 추가?

- a. parameterType = 넘겨받는 파라미터
 - i. 상품 정보를 등록하려면 상품 정보를 넘겨줘야함
- b. resultType 리턴 타입
 - i. list로 바꾸는건 마이바티스가 알아서해줌

```
<insert id="regist" parameterType="productDto">
    insert into product (code, model, price, id)
    values (#{code}, #{model}, #{price}, #{id});
</insert>
```

17.