

sts4 tool	spring boot 가능(spring mvc project 생성불가)
help-eclipse marketplace- sts3 add on 툴 추가 설치(플러그인 기능)	spring mvc project 생성 메뉴--> 4.3.18 업그레이드

## spring mvc project

\*.xml 설정 필요--> annotation 대체 줄일 수 있다

여전히 xml 태그 설정 남아있다.

xml 파일 갯수 많다.

## 32장

## spring boot-2014년

백엔드프로젝트 – spring mvc project

최종프로젝트 – spring boot

## - spring boot project

1> pom.xml 파일 내부 스프링 버전 설정 없다

부트에서 자동 설정한다

2> src/main/java/패키지명.프로젝트명Application.java - main 시작 실행한다

3> @SpringBootApplication 선언한다

4> run as- spring boot app 실행한다

5> tomcat 내장-실행한다

6> 실행 tomcat 포트 8080 동작한다.

7> 포트 수정 가능하다

8> jsp 기본 뷰 아니다 – 설정 추가 필요하다 – controller=model=view

8-1.ng.mvc.view.prefix=/WEB-INF/views/

spring.vc.view.suffix=.jsp

## 8-2. src/main/webapp/폴더 생성

## 8-3. pom.xml

```

<!-- for jsp -->
<dependency>
<groupId>org.apache.tomcat.embed</groupId>
<artifactId>tomcat-embed-jasper</artifactId>
<scope>provided</scope>
</dependency>
<!-- for jsp jstl -->
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
</dependency>
<!--for tomcat restart -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<optional>true</optional>
</dependency>

```

9> src/main/resources – java, jsp 제외 (html xml jpg js css)

**application.properties** 파일 "모든 부트" 설정

10> src/main/resources/static 폴더 -> html jpg js css 저장

application.properties

```
server.port=8081
spring.mvc.view.prefix=/WEB-INF/views/
spring.view.suffix=.jsp
```

- SPRING BOOT PROJECT – MVC, JSP, JQUERY

0>file-new-spring boot-spring starter project 생성

프로젝트명 지정 , packaging-war, java version-8, 패키지명 지정

1>pom.xml

```
<!-- for jsp -->
<dependency>
<groupId>org.apache.tomcat.embed</groupId>
<artifactId>tomcat-embed-jasper</artifactId>
<scope>provided</scope>
</dependency>
<!-- for jsp jstl -->
<dependency>
<groupId>javax.servlet</groupId>
<artifactId>jstl</artifactId>
</dependency>
<!--for tomcat restart -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<optional>true</optional>
</dependency>
```

2>src/main/resources/application.properties 편집

```
#port
server.port=8081

#뷰 jsp
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

# java 변경 재시작
spring.devtools.restart.enabled=true
# jsp 변경 재시작
spring.devtools.livereload.enabled=true
```

3> src/main/resources/static폴더 \*.html, \*.js, image 붙여넣기

4>

src/main/java/패키지명.부트프로젝트명Application.java 찾는다

@SpringBootApplication

@ComponentScan(????)

main

*.java	*.jsp
@Controller class A{ @RequestMapping("/a") sample.jsp 뷰 } ==> <a href="http://localhost:포트명/a">http://localhost:포트명/a</a>	src/main/webapp/폴더 생성 sample.jsp 완료했습니다.

src/main/java/패키지명.부트프로젝트명Application.java – run as – spring boot app 실행한다  
브라우저에서 <http://localhost:포트명/a> 실행해본다

=====

SPRING BOOT – "간편"

-파일업로드

spring mvc 1>POM.xml commons-io-xxx.jar commons-fileupload-xxx.jar 2>servlet-context.xml <bean id="mul" class=".....CommonsMultipartResolver/> 3>c:wupload 4>MultipartFile api	spring boot tomat 내장 기본 파일업로드 jar 내장 3>업로드파일 저장 폴더 설정 4>c-m-v(그대로 사용) 5>UploadController-->upload @CompoentScan(base...=UploadController.class)
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

<웹 정적인 요소 – html js css 이미지파일들 현재 스프링부트프로젝트 내부 위치-별도설정추가필요없다>

url--><http://localhost:8081/images/drinks.jpg>

서버파일경로-->c:\workspace\_spring\myboot1\src\main\resources\static\images\drinks.jpg

<웹 정적인 요소 – html js css 이미지파일들 현재 스프링부트프로젝트 외부 위치-별도설정추가필요있다>

@Configuration

clas ... implements WebMvcConfigurer{....}

c:wupload폴더파일들을 /upload' url 매핑 설정

현재부트프로젝트외부 파일 브라우저 url 접근

서버파일경로-->c:wupload/images/drinks.jpg

url--> <http://localhost:8081/upload/images/drinks.jpg>(설정)

\$.ajax({ url: '/ajax' data: { '요청파라미터명':값 , , , , } type: 'get' success:function(response){} });</pre>	<p>spring boot 자바객체를 json 변환 라이브러리 내장</p> <pre>@Controller class A{ @ResponseBody @RequestMapping("/ajax") public MemberDTO m(){     return new MemberDTO("", ""); }</pre>

#### - mybatis

<p>mbatis-config.xml (이름-수정가능) sql매핑파일.xml</p>	<p>1. a.MemberDTO 클래스 이름 ALIAS="dto" &lt;select resultType=" a.MemberDTO &lt;select resultType=" dto</p> <p>2. data 연결설정 3.sql매핑파일명설정</p>
----------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

#### - mybatis+spring mvc

<p>mybatis-config.xml (이름-수정가능) sql매핑파일.xml + spring-mybatis.xml(이름 수정 가능) 2. data 연결설정 3.sql매핑파일명설정 + web.xml servlet-context.xml</p>	<p>1. a.MemberDTO 클래스 이름 ALIAS="dto" &lt;select resultType=" a.MemberDTO &lt;select resultType=" dto</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------

- mybatis+spring boot

spring boot project 항목 web- spring web 선택-mvc	추가 부트프로젝트-우클릭-spring- add starters sql- jdbc api mybatis api mysql jdbc driver api 선택
mybatis-config.xml (이름-수정가능) + application.properties	1. a.MemberDTO 클래스 이름 ALIAS="dto" <select resultType=" a.MemberDTO <select resultType=" dto
MemberDAO          sql매핑파일.xml	@Mapper @Repository 인터페이스 @Autowired SqlSession seesion;==>삭제  메소드 session.xxxx("")==>선언부만 남김  <mapper namespace="@Mapper선언인터페이스 명" >

## spring boot mybatis

spring.mtybatis.MemberMybatisController	@Controller @ReuestMapping (불변)
spring.mtybatis.MemberService	인터페이스
spring.mtybatis.MemberServiceImpl	@Service("이름")
spring.mtybatis.MemberDAO	@Mapper @Repository("이름 interface MemberDAO 모든 메소드 선언부- 이름은 sql 매핑파일 id 동일해야 한다
spring.mtybatis.MemberDTO	@Component("이름"
스프링부트메인클래스	@StringBootApplication @MapperScan(base.....=MemberDAO.class) @ComponentScan(base.....=MemberDAO.class)
sql-mapping.xml	<mapper namespace="spring.mybatis.MemberDAO" <select id=""

## spring boot mybatis

MYSQL-employeesdb

create table product (

PRODUCTcode int(5) primary key not null,

PRODUCTname varchar(50) not null ,

price int(10) not null,

company varchar(100) ,

balance int(5),

regdate datetime

);

insert into product values(1, '컴퓨터', 100000, '삼성', 3, now());

insert into product values(2, '마우스', 10000, 'LG', 30, now());

insert into product values(3, '모니터', 100000, '애플', 5, now());

insert into product values(4, '세탁기', 900000, '테슬라', 10, now());

insert into product values(5, '냉장고', 900000, '삼성', 6, now());

SELECT \* FROM PRODUCT;

==> 5 조회상태

product.ProductController

@Controller @RequestMapping (불변)

/productlist 매개 변수 없다-product리스트모델-  
mybatis/productlist

1 [컴퓨터](#) 100000

.....

a태그 상품명 클릭하면 /product?code=1

-----

/product?code=1

해당 코드 product 1개 정보-모델 정의-  
mybatis/detailproduct 뷰

코드:1

이름:컴퓨터

가격:100000

제조회사:삼성

재고량:3

입고일:xxxx

product.ProductService

인터페이스 선언메소드

product.ProductServiceImpl

@Service("이름")

product.ProductDAO

@Mapper

	@Repository("이름 interface ProductDAO 모든 메소드 선언부- 이름은 sql 매핑파일 id 동일해야 한다
product.ProductDTO	@Component("이름"
스프링부트메인클래스	@StringBootApplication @MapperScan(base.....=MemberDAO.class) @ComponentScan(base.....=MemberDAO.class) @MapperScan(base.....=ProductDAO.class) @ComponentScan(base.....=ProductDAO.class)
sql-mapping.xml -->member_table 기능 sql  src/main/resources/mybatis/mappers/ product-mapping.xml 새로 생성	<mapper namespace="product.ProductDAO" <select id="??????"
application.properties	#3.mapper 설정 mybatis.mapper- locations=classpath:mybatis/mappers/product- mapping.xml