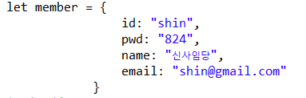
@RequestBody

클라이언트가 보내는 RequestBody를 java 객체로 변환



json 형태의 데이터, ajax로 전달

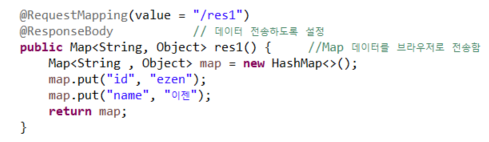


json형태로 받은 Requestbody를 MemberDTO 객체에 저장

@ResponseBody

리턴값을 view를 통해서가 아닌 Http responseBody 즉 요청본문을 써 전달

쓰기 전에 MessageConverter에서 변환이 이뤄진 후 써지게 됨



Map객체를 브라우저에 return



@ResponseBody에 의해 json으로 변환됨

@RestController = @Controller+@ResponseBody

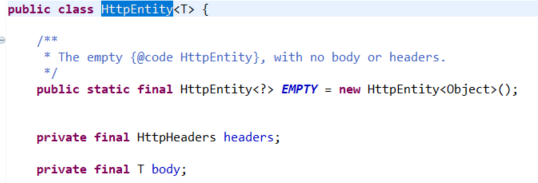
ResponseEntity

반환하는 리소스에 body 뿐만 아니라 header, httpStatus를 담아 전송할 수 있는

클래스



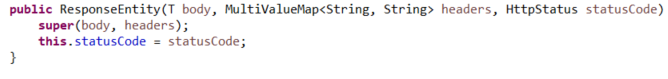
ResponseEntity는 HttpEntity를 상속 받는다



HttpEntity에 지정된 속성들



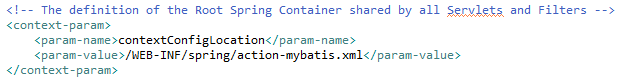
body와 header 정의 메서드



super(body,headers)는 위 메서드를 상속 받는다

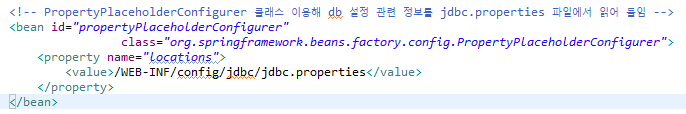
15. 게시판 만들기

JDBC 연결

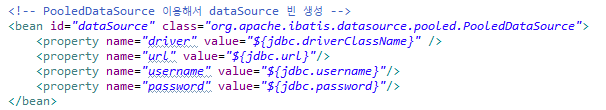


spring 컨테이너 실행시 action-mybatis.xml 전역변수로 선언

action-mybatis.xml

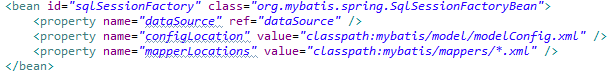


PropertyPlaceholderConfigurer를 통해 jdbc.properties 파일의 db설정 정보를 가져옴



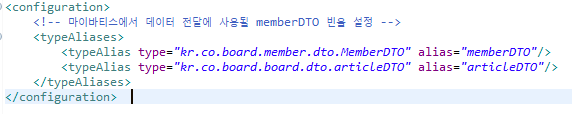
PooledDataSource를 “dataSource” 빈 생성하고 PropertyPlaceholderConfigurer

로 가져온 값들을 입력



SqlSessionFactory를 “sqlSession” 빈으로 생성하고

data Source, moedelConfig(DTO)의 위치, 매퍼의 위치 주입



modelConfig.xml에 mybatis에서 데이터 전달에 사용될 DTO 빈들을 설정한다

\

SqlSessionTemplate을 sqlSession 빈을 생성

tiles 설정



web.xml에 servlet-context를 통해 DispatcherServlet을 설정하도록 선언

 UrlBasedViewResolver를 뷰 리졸버로 사용, tilesview 빈 주입

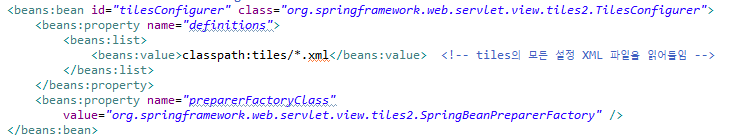
InternalResourceViewResolver

war파일 내에서 접두와 접미를 붙여 뷰 템플릿의 경로를 찾음

UrlBasedViewResolver의 하위 객체

UrlBasedViewResolver

view 이름과 view자원과의 이름이 같을 때 사용할 수 있음



TilesConfigurer 클래스를 이용해 “tilesConfigurer” 빈생성

tiles 관련 설정

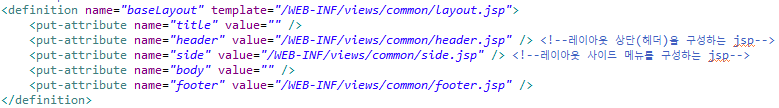
SpringBeanPreparerFactory : 타일즈의 부가적인 기능 구성

tiles.\*xml 설정

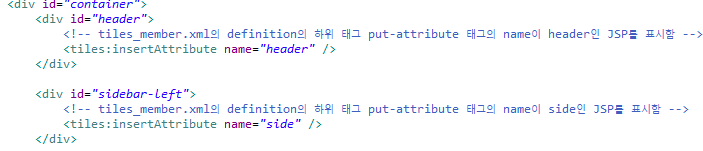


DTD 선언

<tiles-definitions>



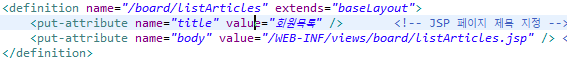
<definition> layout.jsp(template)에 구성 jsp(attribute) 기술



layout.jsp에 <tiles:insertAttribute>에 xml에 <put-attribute>한 name으로 jsp 표시



baseLayout을 상속하여 main.do 요청 시 페이지 설정

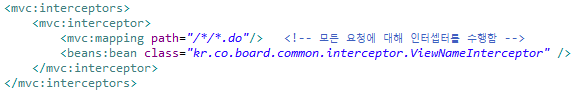


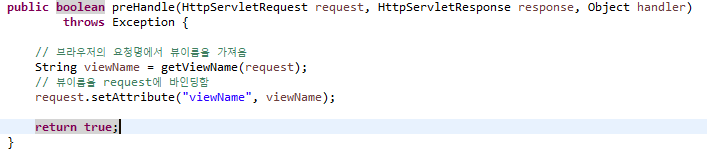
baseLayout을 상속하여 /board/listArticles.do를 요청 시 페이지 설정

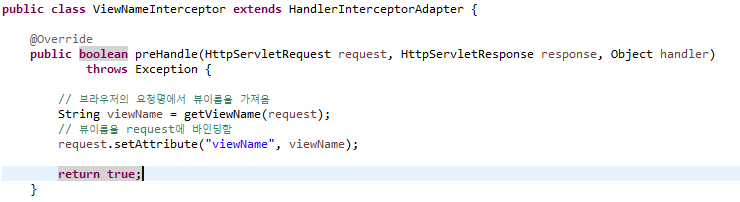
interceptor 설정



Servlet-context에 <mvc:interceptor> 사용을 위한 xmls:mvc 설정







인터셉터로 사용할 클래스는 HandlerInterceptorAdapter 상속해야 함

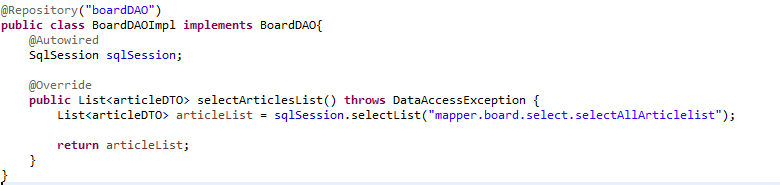
setViewName 메서드를 통해 jsp의 이름을 구해 request에 담아 전송

MVC

ArticleDTO

DB로 사용할 table의 요소를 작성하고 setter, getter와 생성자 메서드를 생성

직접 작성한 클래스 이므로@Component로 빈 등록

BoardDAO 

BoardDAO를 상속 받는 BoardDAOImpl에 @Repository로 빈 등록

action-mybatis.xml에 sqlSession을 빈으로 등록했으므로 @Autowired 가능

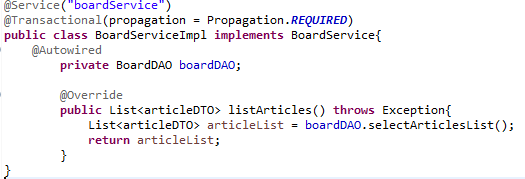




SqlSession의 메서드 selectList로 mapper.board 네임스페이스의 selectAllArticlelist

메서드를 articleDTO에 담아 리턴

BoardService



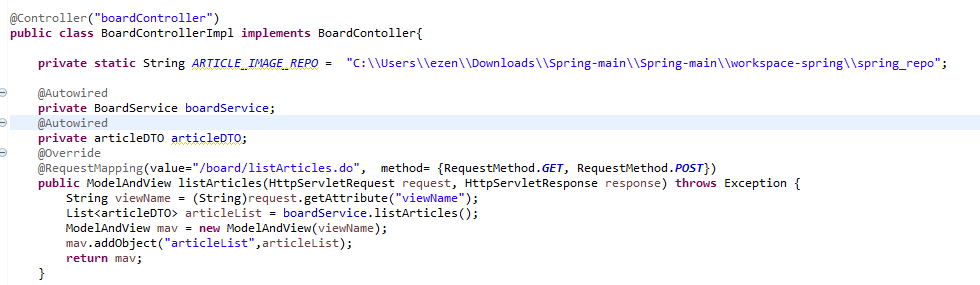
@Service 로 서비스빈으로 등록

서비스클래스 이므로 트랜잭션 적용 가능

@Repsitory로 빈 등록한 BoardDAO를 @Autowired로 불러옴

board의.selectAll Articlelist 메서드 실행

Controller



@Controller로 빈 등록

@Service로 등록한 빈과 setter, constructor로 등록한 @component ArticleDTO 빈을 불러옴

interceptor preHandle메서드로 전달받은 request영역의 viewName을 선언

viewName으로 mav선언

listArticles()메서드로 받아온 articleList 오브젝트를 mav에 추가

s