MYBATIS FRMEWORK

- 객체지향 언어인 자바의 관계형 데이터 베이스 프로그래밍을 좀더 쉽게 할수 있게 도와주는 개발 프레임워크.

- 자바에선 데이터베이스 프로그래밍을 하기 위해 JDBC(자바에서 제공하는 데이터베이스 프로그래밍 API)를 제공

- JDBC는 관계형 데이터 베이스를 사용하기 위해 다양한 API를 제공

- 처음에는 ibatis로 시작

java framework

spring 1> ioc, di, mvc, db 연동, 다른 프레임워크 통합 기능 2-1. spring legacy – spring mvc 2-2. spring boot – 설정 단순화	mybatis 1>db 연동 기능 2> jdbc 대체 기술

java db 연동 종류

jdbc	mybatis	mybatis+spring mvc	기타
jdk설치	jdk설치 +mybatis라이브러리 설치 ==> spring mvc – mvnrepository.com spring boot – 항목 선택 자동 다운로드 가능	jdk설치 +mybatis라이브러리 설치 +spring라이브러리설치	spring jdbc hibernate

java db 연동 코드

jdbc	mybatis	mybatis+spring mvc

class JDBCDAO {
....
Class.forName("jdbc driver명");
Connection conn = DriverManager.getConnection ("jdbc:oracle:thin:@127.0.0.1:1521:xe", "hr", "hr");
PreparedStatement pt = con.prepareStatment("select * from emp where id=?");
ResultSet rs = pt.executeQuery();
rs.next();
....
 	
}
	class MybatisDAO {
	SqlSessionFactoryBuilder builder =
	new SqlSessionFactoryBuilder();
	SqlSessionFactory factory =builder.build (Resources.getResourceAsReader ("mybatis/mybatis-config.xml"));
	SqlSession session = factory.openSession(true);
	
	session.selectOne()
	session.selectList()
	session.insert()
	session.update()
	session.delete()
		@Mapper
		class MybatisSpringDAO {
		@Autowired
		SqlSession session ;
		
		session.selectOne()
		session.selectList()
		session.insert()
		session.update()
		session.delete()

별도 설정 파일 x	mybatis-config.xml sql-mapping.xml (이름 변경 가능)	spring boot 에서는 application.properties

spring boot 에서 mvc + mybatis 사용하기

1. spring starer project 생성
2. starter 항목에서 jdbc api(spring에서 myabatis 연동), mybatis framework api, oracle driver or mysql driver 선택
3. starter 항목에서 spring web 선택- tomcat 실행 필요
4. pom.xml 확인
5. pom.xml에서 jsp 사용 위한 dependency 태그 추가
(spring legacy – 기본 view –jsp)
(spring boot – jsp view 설정 추가)
6. maven dependencies 확인 – 라이브러리 다운로드 확인(*.jar)
7. application.properties
web설정
port

view 경로
view 확장자

mybatis 설정
jdbc driver 명
url
user
password
mybatis-config.xml(마이바티스설정파일)

파일업로드
기본 업로드 파일 크기 1mb 제한 더 큰 크기 설정 변경

oracle – employees 테이블

8. mybatis-config.xml...., sql-mapping.xml
9. MybatisController.java , MybatisService.java MybatisServiceImpl.java, MybatisDAO.java, MybatisVO.java

select 1개 레코드
select 여러개 레코드
insert
delete
update

-spring boot 파일 위치

*.java	src/main/java
*.jsp	src/main/webapp(폴더생성) + application.properties 설정 src/main/webapp/WEB-INF/views/
html xml css js properties	src/main/resources/application.properties src/main/resources/static/*.html, *.css, *.js

	src/main/resources/폴더생성/*.xml src/main/resources/*.xml

product 테이블 – sql command line 또는 sts data source explorer

```
CREATE TABLE product (
                prdNo VARCHAR2(10) NOT NULL PRIMARY KEY,
                prdName VARCHAR2(30) ,
                prdPrice NUMBER(10),
                prdCompany VARCHAR2(50),
                prdStock NUMBER(5)
  );

  INSERT INTO product VALUES('1001', '세탁기', 100000, '삼성', 10);
  INSERT INTO product VALUES('1002', '냉장고', 120000, 'LG', 3);
  INSERT INTO product VALUES('1003', '프린터', 300000, 'HP', 5);

  SELECT * FROM product;
```

product.ProductController	@Controller
product.ProductService	인터페이스
product.ProductServiceImpl	@Service
product.ProductDAO -> <SELECT ID=""	@Repository 클래스—1개 메소드 이름 사용자 정의 – 1개 mapping sql 실행 구현 @Repository @Mapper 인터페이스-1개 메소드 이름 mapping.xml 파일 설정 id값 동일 - 1개 mapping sql 실행(구현x)
product.ProductVO	@Component
스프링부트메인	@MapperScan(base.....=ProductDAO.class)
product-mapping.xml	<mapper namespace="@Mapper인터페이스명" <select id=""

@Service("s") public class ProductServiceImpl implements ProductService { <<spring동작>> ProductServiceImpl s = new ProductServiceImpl(); ProductService productService = new ProductService스프링내부구현클래스();	@Autowired @Qualfier("s") ProductService service;
@Service public class ProductServiceImpl implements ProductService { <<spring동작>> ProductServiceImpl productServiceImpl = new ProductServiceImpl(); ProductService productService = new ProductService스프링내부구현클래스();	
@Service("service") public class ProductServiceImpl implements ProductService { <<spring동작>> ProductServiceImpl service = new ProductServiceImpl(); ProductService productService = new ProductService스프링내부구현클래스();	@Autowired ProductService service; 1> ProductService 타입 bean 객체 조사 2> 2개 이상 중복되면 변수명 일치 객체 주입
@Service("s") public class ProductServiceImpl implements ProductService { <<spring동작>> ProductServiceImpl s = new ProductServiceImpl();	@Autowired @Qualifier("s") ProductService service; 1> ProductService 타입 bean 객체 조사 2> 2개 이상 중복되면 변수명 일치 객체 못찾았다

ProductService productService = new ProductService스프링내부구현클래스();	3> @Qualifier("s") s 이름의 bean 찾아서 service 주입