

- spring boot project

ci cd – 프로젝트 개발 – 변경사항/공유 – 수정 유지

지속적 통합 – 배포

1>git command

spring – 개발방법 다양(xml + @xxxx)

spring mvc project / spring boot project

2>git sts 연동-내일

- spring boot project=spring starter project

1> application.properties 파일 설정

2> jsp 사용 설정 추가 필요하다

pom.xml / application.properties

3> 컨텍스트 없다 . <http://ip:port/xxxx>

4> main 메소드 있다- main클래스 실행- run as - spring boot app

5> 내장 tomcat 동작하다- application.properties 포트설정 변경

6>tomcat 자동재시작 설정 변경

7>브라우저 수동 실행- 주소 입력

8> src/main/resources/application.properties

src/main/resources/static/html css js gif 저장

src/main/resources/templates/ 타임리프 포함 html파일(설정추가)

뷰 – 타임리프(*.html) 기본 템플릿

<th:xxx>

src/main/webapp	reources	html css js gif 저장	
	WEB-INF	web.xml	
		views	jsp포함
		spring	설정xml포함

9> spring mvc – pom.xml 라이브러리 추가

9-1. ajax - boot 라이브러리 이미 포함

@ResponseBody 이용하려면

9-2. file upload – boot 라이브러리 이미 포함

MultipartFile api 이용하려면 ==> CommonsMultipartFileResolver

```
<beans:bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartFileResolver" />
```

==> 자동 포함

- mybatis – spring mvc

1> sql 매핑 xml 단독 설정

--.> dao 클래스 구현 메소드 이름 선언부....

2> sql 매핑 xml + dao 파일들 같이 매핑

- mybatis – spring boot

2> sql 매핑 xml + dao 파일들 같이 매핑

@Mapper @Repository interface EmpDAO List<EmpVO> getEmpList() ;	sql 매핑 xml
@Repository class EmpDAO @Autowired SqlSession session; List<EmpVO> getEmpList(){ session.selectList("emplist") }	<mapper: EmpDAO <select id="getEmpList" resultType="empVO"

The screenshot displays the Spring Tool Suite 4 IDE with two files open: `EmpDAO.java` and `sql-mapping.xml`.

EmpDAO.java (Left Panel):

```
1 package spring_mybatis;
2
3 import java.util.HashMap;
4
5 @Mapper
6 @Repository
7 public interface EmpDAO {
8
9     public List<EmpVO> getEmpList();
10
11     public EmpVO getEmpOne(int id);
12
13     public List<EmpVO> getEmpName(String name);
14
15     public List<EmpVO> getEmpWhere(int id);
16
17     public void insertEmp(EmpVO vo);
18
19     public void updateEmp(EmpVO vo);
20
21     //dao - 매소드 - mapping sql 1개 실행
22     public void deleteHistroy(int id);
23
24     public void deleteEmp(int id);
25
26     public int getEmpCount();
27
28     public List<EmpVO> getEmpPage(int[] ar);
29
30     public void insertEmpWithSeq(EmpVO vo);
31
32     public List<EmpVO> getEmpWithArray(int[] dept);
33
34     public List<EmpVO> getEmpWithList(List<Integer> list);
35
36     public void updateEmpWithMap(HashMap<String, String> map);
37
38 }
```

sql-mapping.xml (Right Panel):

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
3 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4 <mapper namespace="spring_mybatis.EmpDAO">
5
6     <select id="getEmpList" resultType="empVO">
7         select * from employees
8     </select>
9
10    <select id="getEmpOne" resultType="empVO"
11        parameterType="int">
12        select * from employees where employee_id=#{id}
13    </select>
14
15    <select id="getEmpName" resultType="empVO"
16        parameterType="String">
17        select employee_id, first_name, last_name
18        from employees where first_name=#{a}
19    </select>
20
21    <select id="getEmpWhere" resultType="empVO"
22        parameterType="int">
23    <![CDATA[select employee_id, first_name, last_name
24        from employees where employee_id <= #{e}]]>
25    </select>
26
27    <insert id="insertEmp" parameterType="empVO">
28        insert into employees
29        (employee_id, last_name, email, hire_date,
30        job_id, department_id)
31        values
32        (#{employee_id}, #{last_name}, #{email},
33        sysdate, #{job_id}, #{department_id})
34    </insert>
35
36    <update id="updateEmp" parameterType="empVO">
```

The bottom status bar shows the time as 18:54:12 and the date as 2021-04-17.