

CI / CD

CONTINUOUS INTEGRATION

CONTINUOUS DEVELOPMENT , DEPLOY

지속적인 통합 / 지속적인 개발, 배포

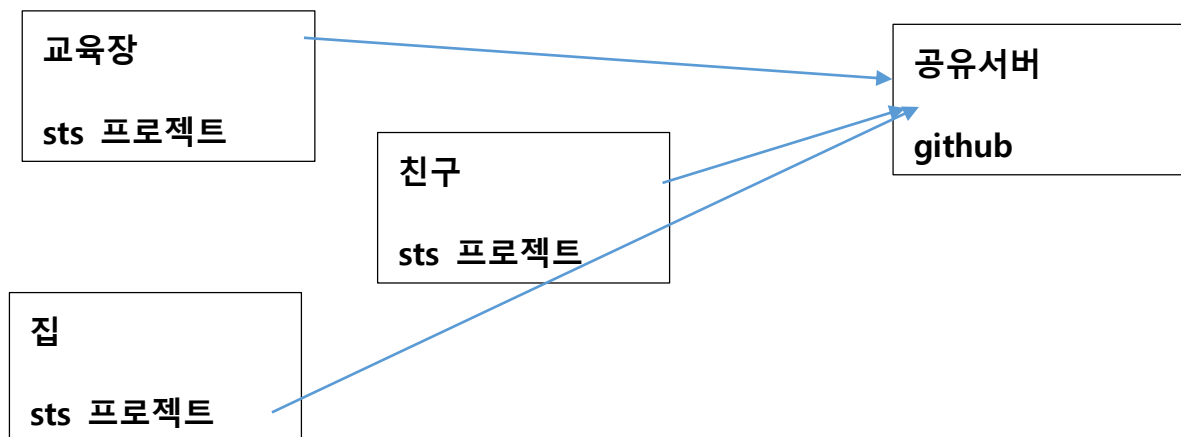
==> 프로젝트 협업 - 각자 개발 / 통합 / 수정 / 반영 / 재통합 / 수정 / 반영 / 재통합 / .....-->"공유"

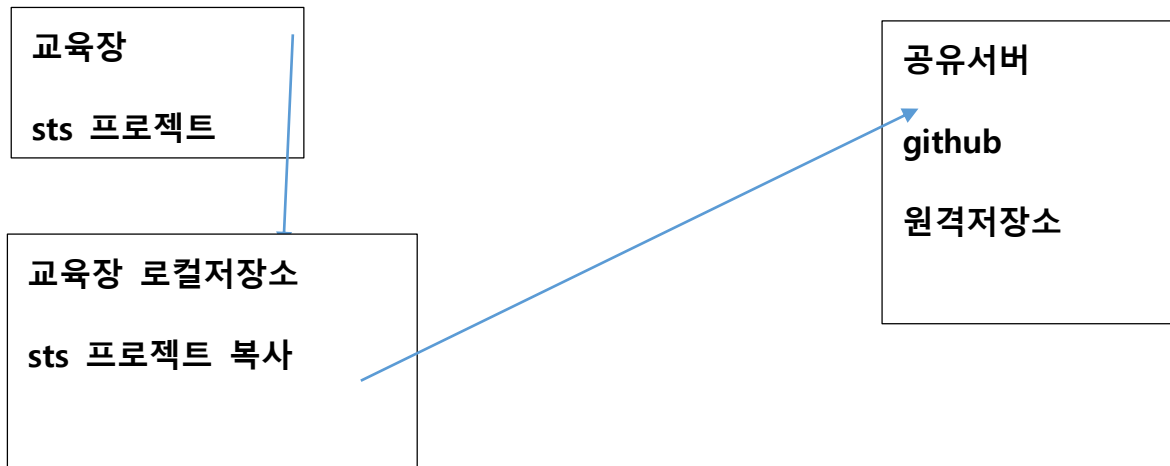
형상관리-버전관리 =>

SVN - > github

2일 특강 -> 코맨드 형태

sts / eclipse + github ---> "spring project"





원격저장소-remote repository(원격url)

로컬저장소 – local repository(내컴퓨터 폴더 지정)

commit – 로컬 저장소에 내 프로젝트를 저장

push – 원격 저장소에 내 프로젝트 전송(upstream)

fetch – 원격 저장소에서 프로젝트를 가져오기

merge – 기존 내 프로젝트와 fetch한 프로젝트 병합하기

pull – fetch + merge

## github와 sts 또는 eclipse 연동

-github 회원가입과 원격저장소 생성-

원격저장소 <http://github/xxxx/repository/>

로컬저장소

이클립스용 - c:/ kdigital/eclipse/git/레포지토리

sts용 - - c:/ kdigital/git/레포지토리

연동

git repository perspective 연다

(프로젝트 없는 상태)

-내 프로젝트 원격저장소에 올리기-

1. 프로젝트 생성
2. team-share project
3. team-add to index
4. team-commit...
5. commit메시지 입력하고 push and commit 클릭
6. 로컬 저장소 확인

git repository perspective 연다

(working tree - 프로젝트 보이는 상태)

7. 원격 저장소 확인

<http://github/xxxx/repository/> --> 프로젝트 보이는 상태

-원격저장소에서 프로젝트 가져오기-

1. git repository perspective 연다
2. import project
3. github 이메일/암호

## -변경 사항 만들기-

1. 이클립스에서 프로젝트\src\main\java\com.xxxxx.HelloController3.java 추가

team – add to index

team – commit...

2. 로컬 저장소와 원격 저장소 저장-수정 반영

3. 이클립스의 로컬 저장소와 원격 저장소 확인

4. sts 추가된 사항 가져오기

team-pull

(sts – 원격저장소 한번 입력, 아이디, 암호 입력 – 저장 )

(sts – 원격저장소 삭제 – sts 재시작)

## -github 다른 계정 프로젝트를 내 레포지토리로 작업 독립적 –

1> 팀장 레포지토리를 각 팀원들 레포지토리 가져온다

2> 팀장 레포지토리랑 별도로 팀원들의 수정 삭제 작업을 한다

3> 팀장에게 수정 삭제 합병 최종 반영 허락받는다

4> 팀장이 최종적 합병한다

5> 팀원 1개 레포지토리 공유 작업 동시성 문제 문제-

팀장-레포지토리 합병	팀원1-레포지포리 복사-작업-알려준다
1>팀원들이 공유한 레포지포리 생성	팀원2-레포지포리 복사
2>fork - pull request	팀원3-레포지포리 복사

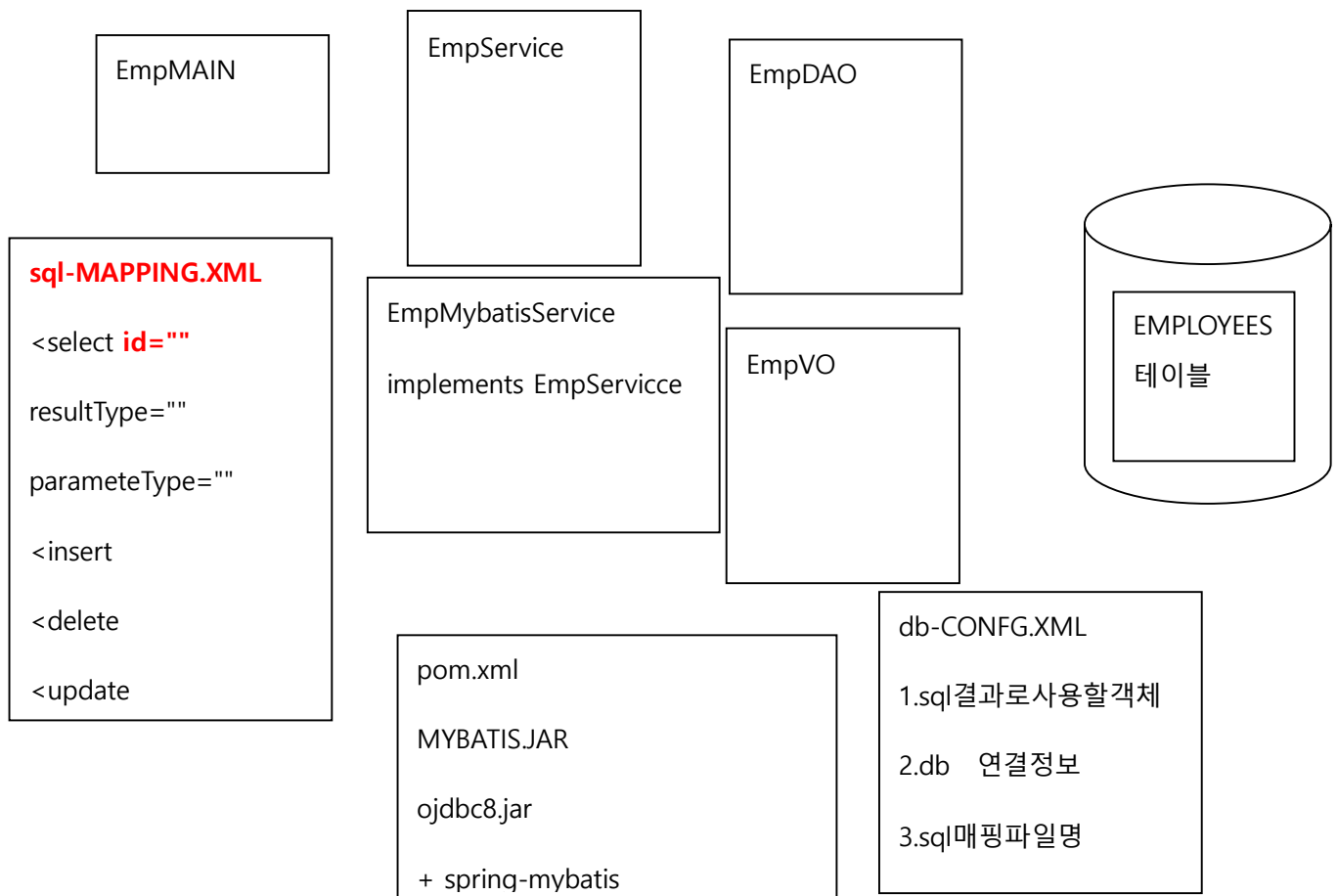
팀장

팀원

허진호	김다현
김세미	김승겸
김승현	김우일
김혜린	김환준
문병전	박건우
박소연	정동현
서형준	안재완
이민정	전영빈
이현용	하은

1. 깃헙 새로운 레포지토리 생성 2. sts 프로젝트 생성 3. 원격저장소로 업로드 18. pull request 내용 확인한다. 19.merge 한다. 20. 팀원에게 알려준다.	4. 팀장이름/레포지토리 검색 5. fork 6. main 선택하여 새로운 브랜치를 생성한다. 7. 새로운 브랜치는 로컬 저장소에 clone 한다. 8. import project 한다. 9. 변경사항 발생시킨다 10.team – add to index 11.team-commit... 12.commit 누른다 13.로컬 저장소 변경 확인한다 14. team-push 새로운 브랜치명 15. 원격 저장소 변경 확인한다 16. pull request 한다 17. 메일... 21. 현재 fork 레포지토리 삭제한다

## - mybatis



## -mybatis + spring main

EmpMAIN  
->스프링설정파일

EmpService

**@Repository**  
EmpDAO  
소스

sql-MAPPING.XML  
  
<select id=""  
resultType=""  
parameteType=""  
  
<insert  
  
<delete  
  
<update

**@Service**  
EmpMybatisService  
implements EmpService  
  
**@Autowired**  
EmpDAO dao;

**@Component**  
EmpVO



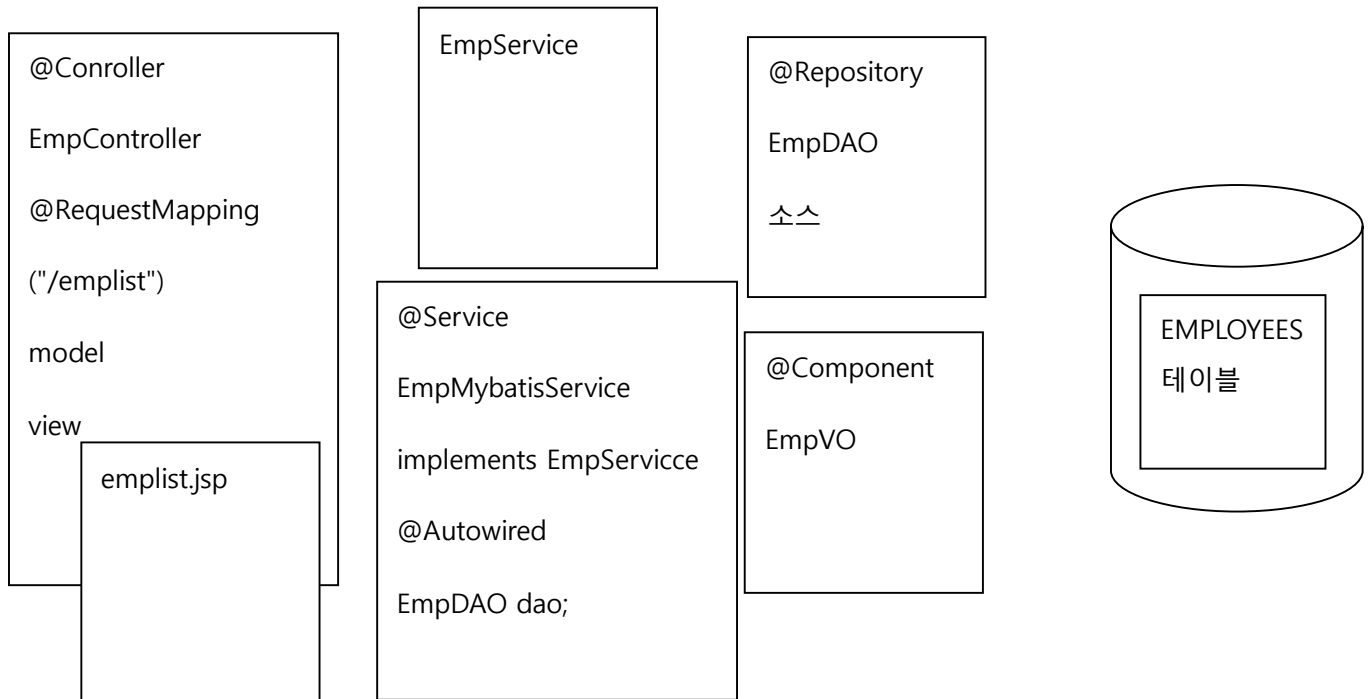
pom.xml  
  
MYBATIS.JAR  
  
ojdbc8.jar  
  
+ mybatis-spring  
  
- spring-jdbc

스프링설정파일.XML  
  
1.db 연결정보  
  
2. 마이바티스 db-  
config.xml 존재  
  
3.sql매핑파일명 존재  
  
4. ANNOTATION 인식  
태그

db-CONFIG.XML  
  
1.sql결과로사용할객체



## -mybatis + spring mvc



WEB.XML

1> "/" -DISPATCHERSERVLET  
(SERVLET-CONTEXT.XML 읽으시오)

2>SPRING MVC +  
다른 연동 XML

**스프링설정파일.XML**

스프링설정파일.XML

1.db 연결정보

2. 마이바티스 db-config.xml 존재

3.sql매핑파일명 존재

4. ANNOTATION 인식 태그

sql-MAPPING.XML

<select id=""

resultType=""

parameteType=""

<insert

<delete

<update

db-CONFIG.XML

1.sql결과로사용할객체

SERVLET-CONTEXT.XML

**4. ANNOTATION 인식 태그**

5 시 SERVLET JSP 스프링 시험  
 과제물 제출-러닝 클라우드  
 GITHUB -> 과제 URL 접속 불가-확인중

23 장 마이바티스  
 24 장 마이바티스 + 스프링  
 25 장 트랜잭션 처리 제외  
 26 장 @  
 27 장  
 스프링프로젝트

1> 수동 LIBRARY(\*.jar) 복사  
 2> sts 설치  
 3> 32 장 스프링 부트

28 장  
 jsp, spring(다중)  
 - 파일 업로드(클라이언트가 서버로 )  
 <input type=text password  
 <textarea  
 <select  
 ==>문자데이터 전송

| client                      | server   |
|-----------------------------|--|
| 1> 파일은 데이터 크기가 큰 경우가 많을 것이다 | pom.xml<br>commons-fileupload.jar<br>commons-io.jar  |
| 2> form method=post 여야 한다   | servlet-context.xml<br><pre> &lt;beans:bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver" /&gt; </pre> |

3>                    form enctype=multipart/ form-data 4> <b>input type=file</b> 이어야 한다(파일명 +파일내용선택)  input type=text id=xxx ===== ===== pw=xxxx <hr/>	<b>MultipartFile</b> 이용 (파일명, 내용, 길이, 타입....) 특정 서버 경로 지정 (c:\Wupload) 저장 구현  String
--	---

UploadVO

UploadController – 2 개 매핑

uploadform.jsp

uploadresult.jsp

## 29 장

1>http 요청 있으면 서버 응답한다.

2>요청 - 응답 종료 이후 클라이언트 정보 삭제된다

3> 세션이나 쿠키 이용하면 이전 클라이언트 정보 유지한다

4> 요청 1 - 응답 1(html 태그 포함 화면 새롭게 구성)

-> 요청 2-응답 2(html 태그 포함 화면 새롭게 구성)

==> 동기화 통신 방식(synchronous)

요청 1 - 응답 1 - 요청 2-응답 2

요청 1 - 요청 2 - 응답 2 - 응답 1

view 화면

```
$("#div").html(" RESULT 변수값  
")  
  
아이디  
  
암호  
  
로그인버튼  
  
<div>?????</div>
```

```
서버  
  
@Controller  
  
...  
  
return  
  
"{W"RESULTW" : W"로그인정상W"}";  
  
결과
```

- ajax

asynchronous java script and xml

화면 구성 + 서버 요청 처리 결과를 이미 구성 화면 변경

응답 + 요청 + 결과 화면 동적 변경

JSON

- 자바스크립트 객체

```
{id: 'java' , pw: 1234 , name : '' };
```

==> 다른 언어 호환성

```
JSON - {"id": "java" , "pw": 1234 , "name" : "" };
```

java script, html = "", ""  
 java(servlet, jsp, spring) = ""

CLIENT	server spring mvc(jsp 포함) servlet jsp node js.....
html-ajax 요청방식 x javascript/jquery-ajax 요청 \$.ajax({.....});	json 인식 적용 라이브러리 설치 pom.xml = jackson-data-bind.jar  EmpVO-자바객체 JSON 자동 변경  @ResponseBody @RequestBody
r	e

loginajax.jsp  
 LoginAjaxController

<a href="url"  
 <form action="url"  
 \$.ajax()  
 url:  
 json 데이터 리턴

### 32 장 SPRING BOOT 맛보기

1> spring library \*.jar 파일 수십개 프로젝트 복사 저장  
mybatis.jar  
spring-mvc-web.jar  
+.....

2> maven – 스프링 라이브러리 자동 다운로드 기능 포함 프로젝트  
sts  
spring mvc project(=spring legacy project)-오라클, mysql +mybatis 설정파일  
\*.xml 설정  
==> 3 일 세미프로젝트

3> spring boot project(=spring starter project)  
sts

spring boot  
-설정파일(xml 줄이자)  
- db, mybatis, web 체크-자동 설정