

| 웹클라이언트 | 웹서버=자바언어구현 |
|--|--|
| html – 화면구성 css –크기 색상 모양 js jquery – 동작 (화면 변경) | servlet jsp 웹서버 cgi – 클라이언트 수가 늘어나면 서버 프로세스 수 증가한다. ==> 자바 서블릿 jsp "멀티스레드" 다수 클라이언트 처리 동시 |

| servlet | jsp |
|---------------------------|---|
| 자바 언어 구현 + html태그 응답 | 자바 언어 구현 + html태그 응답 |
| 자바 클래스 상속 메소드 오버라이딩 | html파일 유사형식 클래스 , 상속 오버라이딩 구현 필요없다(내부적 클래스 , 상속 |

| | |
|---|---|
| doGet / doPost / service 자바문장 내부에 html;태그 포함 String 타입 저장 | 오버라이딩 자동 변경) html파일 내부 태그 양식으로 자 바 문장 포함 형식 |
|---|---|

| html 태그 | html태그 그대로 |
|--|---|
| jsp 태그 <% %> <jsp: /> <c: /> \${xxxx } 내장객체제공 | <% 스크립트릿 태그 %> <%@ 지시자 page include %> <%!멤버변수나 메소드 선언 %> <%=출력, 속성값 표현 %> <%-- 주석 --%> |
| 액션태그-자바문장변경실행 "동작" | <jsp:xxxx /> <jsp:forward <jsp:include <jsp:param <jsp:useBean <jsp:setProperty <jsp:getProperty html form 입력 - dto 저장 |
| el | \${el언어} |
| jstl | <c:xxx /> |
| 내장객체 | jsp - 자바소스변경 _jspService 메소드 내부 지역변수 request response out excetption |

| | |
|--|---------|
| | session |
|--|---------|

14장 EL / JSTL

서블릿 응답 표현 html태그를 String,...

- > jsp : html + <% 자바문장 %>
- > 내장객체
- > < jsp:include
- > expression language(jsp파일내부) + jsp tag library

표현언어

- 객체공유

| jsp | el |
|---|-------------------------|
| <% request.setAttribute("이름", 객체) (타입)request.getAttribute("") -> 이동 , 포함되는 파일까지 | \${requestScope.이름} |
| <% session.setAttribute("이름", 객체) (타입)session.getAttribute("") -> 브라우저 실행 모든 파일 | \${sessionScope.이름} |
| <% application.setAttribute("이름", 객체) | \${applicationScope.이름} |

| | |
|---|--|
| (타 입)application.getAttribute("") -> tomcat 서버 같은 웹프로젝트 모든 파일 (ServletContext 이름 api) | |
|---|--|

EL 전용 사용 태그들==> jstl – core 태그부분

1> jstl core 라이브러리 설치

| | |
|------------|---|
| ojdbc6.jar | jre system library main + servlet jsp |
| jstl-.jar | 현재 다이나믹 웹프로젝트만 webcontent\web-inf\lib\jstl- 1.2.jar |
| | 모든 dynamic web project tomcat library |

2> jsp파일마다

<%@ taglib prefix="c" uri=<http://java.sun.com/jsp/jstl/core> %>

<c:xxxx

<c:set var="" value="<%=자바변수 %>" />

<c:out value="" />

<c:remove var="" />

554p

`${ a>b? "" : ""}`

jstl

el 조건문

`<c:if test="${1 > 0}" >`

조건만족시 수행 내용 정의

`</c:if>`

`<c:if test="${1 <= 0}" >`

조건만족시 수행 내용 정의

`</c:if>`

`<c:choose-->switch`

`<c:choose >`

`<c:when test="${조건1}">`

만족 수행

`</c:when>`

`<c:when test="${조건2}">`

만족 수행

`</c:when>`

`<c:when test="${조건3}">`

만족 수행

`</c:when>`

`</c:choose>`

jstltest2.jsp?id=jsp&age=15

jstltest2.jsp?age=15===>id 입력 필수

jstltest2.jsp?id=jsp&age=12 -> 초등학생입니다(13세이전)

jstltest2.jsp?id=jsp&age=15 -> 중학생입니다(16세이전)

jstltest2.jsp?id=jsp&age=19 -> 고등학생입니다

jstltest2.jsp?id=jsp&age=20 -> 성인 인증되었습니다.,

ArrayList<BoardDTO>/ 배열/MAP

el 반복문

<c:forEach begin="1" step="1" end="10" >

수행내용

</c:forEach>

<c:forEach items="배열 리스트 맵" var="i" >

 \${i}

</c:forEach>

<c:forTokens

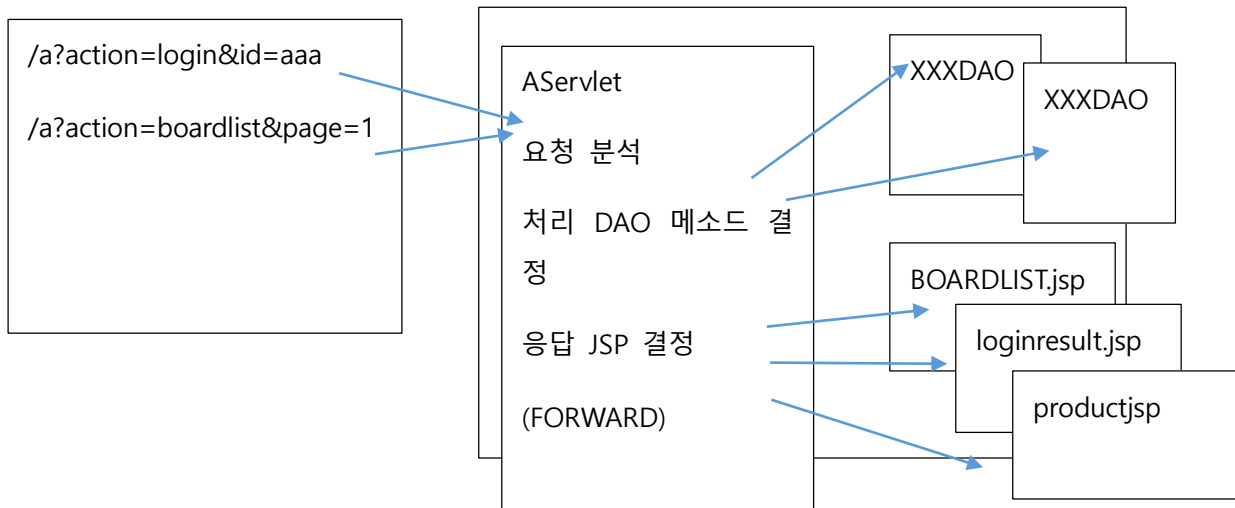
15장 jsp 오픈 소스

1>파일 업로드->spring 파일업로드

16장 html jquery

17장 모델 2 방식 개발- spring

웹서버 개발 형태=형식 = 모델1 방식=흐름 복잡하다



BoardServlet / boardlist.jsp / BoardDAO

BoardDTO

요청 파라미터 추출 - 분석 - 요청 따른 처리 -
결과 응답

서블릿 - controller

xxxdao, xxxDTO - model

JSP - VIEW

MVC방식 구현= MODEL 2

SERVLET + JSP MVC = 권고

SPRING MVC = 강제

18장 스프링

환경설정 복잡

개념, 용어 어렵다

spring framework =

반제품(내 + SRPING 제공) 구현 개발 + 규칙 + 나머지만 내가 개발 = 작업 자동화 도움

프로그램(다수 협업-규칙 약속) 구현 개발 = 기존 개발 수정 단축
기존 개발 수정 --->

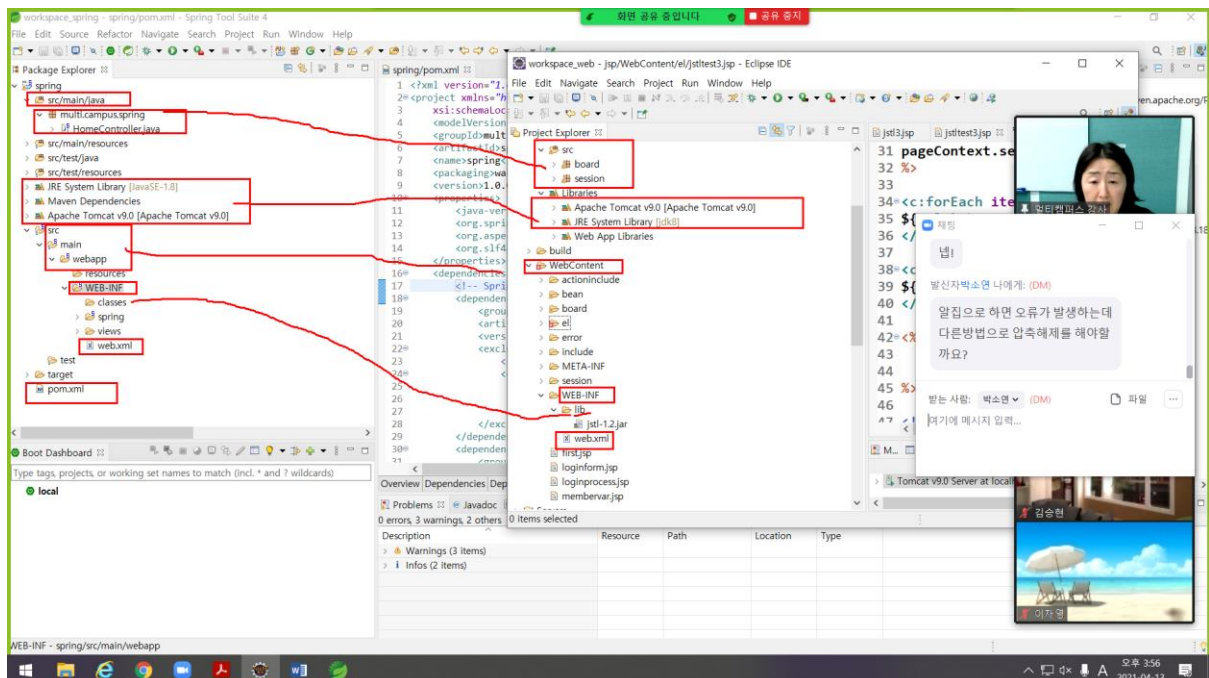
BoardDAO 게시물 관련 기능 메소드 구현 클래스

+

MAIN / SERVLET

틀 정형 프로그램 패턴 (구현 이미-파라미터) + ????

=>



| | |
|--|--|
| interface TV void powerOn(); void powerOff(); void soundUp(); void soundDown(); | |
| class SamsungTV implements TV void powerOn(){ } void powerOff(){} void soundUp(){} void soundDown(){} | class LGTV implements TV void powerOn(){.....} void powerOff(){} void soundUp(){} void soundDown(){} |
| TV tv = new LGTV() | |

| | |
|---|--|
| interface BoardDAOInter { } | |
| class BoardDAOOracle implements BoardDAOInter { 구현-오라클 저장 / 조회 } | class BoardDAOMySQL implements BoardDAOInter { 구현-mYSQL 저장 / 조회 } |

자바 일반 DI ==> SPRING DI + IOC

자바웹 MVC ==> SPRING MVC 강제

JDBC ==> SPRING MYBATIS 연동

=====

파일업로드 AJAX REST

SPRING BOOT

