55일차 강의

* 프로젝트 로드맵
  + 제안서
  + UML, 시나리오 작성, 이벤트 흐름, 사용자가 어디까지 접근이 가능한지 여부
* 톰캣을 내려야하는 경우
* 물리적 변화가 일어날 때
* 폴더나 파일을 만들거나 삭제 할 때
* JSP 파일은 수행될 때 JSP 파일로서 수행되는 것이 아니라, 자바파일로 변환시켜서 수행된다
* 이를 서블릿화라고 부른다
* hello.Jsp 파일이면 -> hello\_jsp 라고 (jsp파일명에 밑줄을 붙인 후) 클래스명을 만들어 자바파일로 변환해준다
* jsp 만들면 자동으로 등록해주므로 우리는 만드는건 상관없음
* 우리가 신경써야하는 것은 Service 뿐이다
* Jsp 파일안에 작성했던, 자바문법는 -> 스크립틀릿에 자바문법 그상태 그대로 적힘.
* 예외발생시 (오류가 떳을 때) 보여질 페이지 만들기

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

// e.printStackTrace();

// 예외처리 페이지로 위엄

req.setAttribute("error", e);

RequestDispatcher dispatcher =

req.getRequestDispatcher("/error.jsp");

* dispatcher.forward(req, res);
* Forwording 방식 (다른페이지로 위임하는 방식 – 객체를 포함하고)
* Lib 아래의 ojdbc6.jar 파일을 삭제하여 오류를 억지로 만든후 테스트해보자
* 공통단 (common)
* 전반적인 업무를 파악하는 프로젝트 리더 한명이 이 공통단 로직을 맡게된다. (가장 실력있는 사람이 하면 된다)
* 이외의 팀은 만들어진 공통단을 사용할수 있으면 된다.
* 변화하지 않는 공통단
* 레이아웃이나 설계의 영역 (기능의 분리)
* <jsp: include /> 를 사용하여 우리회사의 로고나

: header 와 footer의 같은 공통의 내용을 유지할수 있다

* <jsp: include /> 인클루드를 쓰는 이유?
* <jsp:include page=*"/Header.jsp"*/>
* 다른 페이지에 있는 기술을 가져다 쓴다
* 수행의 원리?

(필요할지 안 할지를 판단하고 사용하는 것이다)

* Forward 와 include의 차이점
  + 인클루딩 방식

: 작업을 위임하면 다시 이전 서블릿으로 제어권을 반환한다

: 내 리스트의 일부분으로서 내 안에 포함시켜서 보여주겠다는 것

* <jsp:include page=*"/Tail.jsp"*/>

: 인클루드 방식은 이전의 파일이 그대로 저장된 상태로,

이전 파일도 보여주면서 Tail의 내용을 보여주는 것

: 다양한 파일을 보여줄수 있음 -> 추가되어 하나의 화면 같은 느낌으로

보여진다

* + 포워드 방식

: 작업을 한번 위임하면 다시 이전 서블릿으로 제어권이 돌아오지 않는다

* <jsp:forward page=*"/Tail.jsp"*/>

: 포워드 방식은 이전 파일이 저장되지 않은 상태로.

컨트롤러를 거치지 않고 바로 Tail 페이지의 내용을 사용자에게 전달한다

* + Redirect 방식

: 포워드 방식과 유사하나,

리퀘스트 객체에 담을 내용이 하나도 없다

* <jsp: include /> 를 사용하여

: Header(Top)과 Footer(Bottom) 을 추가함

* 이제 HTML 파일을 만들 이유가 없다
* 모두 JSP 파일 안에서 만들어서 가져오면 그만이다
* JSP도 자바고, 자바도 자바다

: 단지 그 안에서 내용의 표현식이 바뀔뿐

JSP 파일로 작성해도 자동으로 자바파일로 변환시켜 보여짐.

* 신규 회원 가입시 (hmUser 에서 변경할 것)

1. 문제가 발생하면 에러 페이지 호출
2. 에러 페이지가 나오면

페이지 top

페이지 bottom 표시

1. 신규 회원 페이지에

뒤로가기 버튼 작성

* 추가버튼 누르면 – error.jsp 가 뜨드록 만들고,

예외처리된 페이지에는 -> 공통단 header, footer 도 뜨드록

취소버튼 누르면 – 뒤로 돌아가기 되도록 (list 보여주는 페이지로)

* 컨트롤러는 뒤에서 내용에 대한 처리하는 것(보이지않는 영역에서)
* 그리고 나서 컨트롤러 뒤에 화면을 보여주기위해 jsp 파일을 호출한다
* 무조건 반응을 일으키면 (예를 들어 이벤트를 발생시키는 추가나 취소버튼을 누르면 컨트롤러로 연결한다)
* doGet doPost 에서 모델을 처리하고 있다
* 나와 관련있는 것들로 이동할 때 상대경로 표시한다
* 공통단 같은 것은 절대 경로로 표시한다.
* 그림 그려 설명하라 (아이패드에 설명있음)

