step2 2주차 (순서를 외울 것)

- 오늘은 tiles와 svn을 진행할 것

- tiles.

: 타일즈란 라이브러리 인데 메인화면은 총 6개의 jsp파일이 붙어서 하나의 화면으로 나오는 구조로 저번시간에 컨트롤러에서 호출한 main.jsp는 가운데의 contents.jsp 를 의미하는 것

화면의 두개의 예시를 비고해 봤을 때 화면상에 안나오는 파일은 header.jsp 는 화면에 나타나지 않음

HTML의 head tag는 페이지의 정보를 나타내는 역할을 하는데 그 글자들은 우리가 화면을 봤을때 보이지 않음 // 이처럼 header.jsp도 있지만 눈에 보이지 않는 것.

- 해당화면을 만들기 위해서는 tiles라는 라이브러리를 활용했고 어떻게 만들었는지 해볼 예정

- 자바쪽을 만질때는 서버를 항상 끄도록 한다.

- 프로젝트 생성

: 프로젝트명을 s2213 으로 생성 (프로젝트 명의 첫글자는 소문자)

generate example 을 체크하면 최소한의 파일들을 참조할 수 있음 > finish 버튼

- 패키지 익스플로어에 가서 maven dependencies를 가서 보면 ~.jar 는 자바만 읽어들일 수 있는 압축 파일로 해당 파일들은 중간코드 즉, 클래스 파일을 압축해놓은 것으로 이 jar들이 maven에 들어가 있음 maven은 네비게이터에서는 보이지 않기 때문에 패키지익스플로러에 들어가야 보임.

이 maven은 여기에 있는 압축 파일들을 단 하나의 xml파일로 관리하는 것을 말한다.

- 모든 프로젝트 마지막에는 pom.xml이 있는데 거기에 들어가면 맨 처음화면은 관심없고

pom.xml탭을 눌러 코드를 본다. 우리는 first프로젝트와 비교해 가면서 할것임

내가 칠 프로젝트는 왼쪽 // 참고할 프로젝트는 오른쪽으로 두고 작업한다.

우리가 봐야 할 부분은 properties로 이 부분은 우리가 지금부터 타일즈 세팅을 할건데 jar가 필요하다. 아까 말했던 하나의 xml이 바로 pom.xml로 여기에서 수정하게 되면 모든 프로젝트에 적용되는 것

dependency 하나당 아까 처음의 jar파일이 하나에서 네개까지 저장이 되는 것

first에 있는 dependency 2개를 복사 붙여넣기 하면 내 프로젝트에 tiles로 시작하는 jar 파일들이 생성됨 ---- pom.xml로 관리한다는 걸 알수 있음

:: 이러한 행동을 의존성주입, 영어로는 인젝션 한다고 표현한다.

2단계 시작

- dispatcher-servlet

: webapp안에 dispatcher-servlet.xml은 확장자가 xml임에도 불구하고 webapp안에 들어가 있는 이유는 화면과 컨트롤러 관련된 설명을 적어놓는 파일이기 때문이 webapp에 있는 것

우리가 지금 할 타일즈설정은 여기 dispatcher에서 설정해야 적용이 되는 것.

창을 두개 킨 상태에서

퍼스트의 64~76에서 타일즈 뷰 설정 bean이 있는데 이를 복사해서 붙여넣기 (저장 x)

설명 : 이게 무엇이냐 하면 우리가 설정할 것 중에 컨트롤러에 main/main.tiles에서 tiles를 무시하라고 했는데 .jsp 도 붙이지 않았었는데 왜냐하면

p:prefix="/WEB-INF/jsp/egovframework/example/" p:suffix=".jsp"

왜냐하면 여기서 jsp 기본 경로를 지정해 놓았기 때문에 main/mian만 써도 뒤에 .jsp로 인식됨

<bean class="org.springframework.web.servlet.view.UrlBasedViewResolver" p:order="1"

컨트롤러 리턴의 우선순위를 지정하는 것으로 .tiles가 1로 지정되어 있으므로 2로 수정한다.

아까 붙여넣은 두개의 빈 중에 하나를 보면 타일즈 레이아웃 설정을 보면 디스페쳐 서블릿은 타일즈 세팅이 되어있는 파일을 가리키는 것이다. 진짜 타일즈 세팅은

<value>/WEB-INF/tiles/default-layout.xml</value> 여기서 하는 것으로 first 프로젝트에서 복사 붙여넣기 한다.

- default-layout.xml 파일을 연다

: layout.jsp 라는 큰 판이 있고 그 위에 헤더 등등을 올리는 것

주소를 다 작성하면 길기 때문에 이름을 지정하여 헤더면 헤더 등등으로 지정한다.

이 중에

<put-attribute name="content" value="" />는 주소가 지정되어 있지 않음

왜? 안바뀌는 애들은 파일을 지정한것이고, content 같은 경우에는 화면이 계속해서 바뀌기 때문에 어떤 특정 파일을 지정해 줄 수 없기 때문에 동적으로 바뀌는 것이라 따로 아래에 빼줌

경로만 지정한 것 뿐만 아니라 뒤에 .tiles를 적으라고 정의해줌 ("\*/\*.tiles")

타일즈를 적지 않으면 우리가 아까 복사했던 타일즈 뷰 설정을 가져오는게 아니라 다른 2순위의 정보를 대려오게 된다.

왜 디자인도 안나오나? css는 html의 헤드부분에 있는데 지금 contents 부분만 불러오니 헤더를 불러올 수 없어 css도 안나오는 것

first에 있는 파일 복사 붙혀넣기 (egovframework/example/cmmn 폴더)

- default-layout.jsp 를 열어서 (.xml과 다르다 주의할 것)

head 안에 태그를 통해 디폴트헤더.jsp가 들어가고

아래쪽 left 도 동일하게 디폴트레프트.jsp가 들어가는데

이를 위해 위에서 두번쨋 줄 설정을 해주는 이유

- 타일 명 변경시 교체해야 할 부분은

자바 메인업무폴더의 메인컨트롤러의 return 값을 수정 //

jsp default-layout 에서 definition name = “\*/\*. tiles를 수정

3교시 svn 수업시작

- 절대 숙제를 수업시간에 진행한 프로젝트가 아닌 새로운 프로젝트를 생성해서 진행

- svn란

: 통째로 받는것…

지금부터 접속을 할건데

프로젝트 우클릭 > 팀 > 쉐어 프로젝트 > 새로운 경로는 첫번쩨 체크 등등

: 내용을 수정하고나서 저장하면 폴더와 파일에 꺽쇠’>’가 표시되는데 지금 현재

svn에 있는 내용과 차이가 있음을 확인 해줌

- 커밋과 업데이트

: 네가지 통째로 올리기, 통째로 내려받기

일부분 올리기, 일부분 내려받기

혹시라도 svn에 target이라는 폴더가 있으면 건들이면 안된다.

override and update는 상대방의 파일을 받아 들이는 것.

override and commit는 절대로 눌러서는 안되는 버튼 !!!!

내 코드로 전체코드를 바꿔버리게 됨

충돌이 일어났을 땐 해당 코드 내용을 보고 먼저 업데이트 이후 커밋 하는게 나음

- svn

버전관리 시스템으로 커밋을 할때마다 숫자가 하나씩 올라가게 되는데 이는 히스토리내역으로 확인할 수 있음.

* 타일즈 셋팅은 어떻게 하는거에요?

1. 먼저 타일즈 셋팅은 하나의 웹 화면에 여러 영역을 붙여주는 라이브러리로

프로젝트를 생성하고 나서 pom.xml에 들어가 타일즈 버전과 타일즈 라이브러리를 설정함

2. 이후 WEB-INF의 dipatcer-servlet 에들어가 타일즈 뷰와 레이아웃을 설정하고 우선순위를

수정한다.

3. jsp 폴더안에 tiles 폴더를 생성하고 default-layout.xml에 들어가 difinition name=”\*/\*.tiles” 를 확인하고 jsp의 main업무 폴더 및 cmmn 업무 폴더를 수정합니다.

4. 이후 java폴더 들어가 main 업무 폴더의 maincontroller를 들어가서 @RequestMapping의 value값을 확인하고 method 안의 return값이 jsp업무폴더/jsp업무파일명.tiles이름 으로 구성되어 있는지 확인한다.

5. 마지막으로 index.jsp 에 들어가 이전의 @RequsetMapping의 value값과 jsp:forward의 값과 일치하게 만들어주고 톰켓 서버에 올리면 만들어짐

-- 숙제 --

: 프로젝트를 생성하고 타일즈세팅을 하고 (여기는 무엇, 여기는 무엇)

초시계 키고 하는데 5분안으로 셋팅하고 증명할 것 ( 5분이 넘으면 이유를 적을 것 )

5분안으로 들어오는 것에 대한 기록을 적을 것 ( 싫어하는 것 : 프로젝트명 컨셉으로 )

업무폴더 명도 바꿀 수 있으니 꼭 메인이 아니라 다른 것 ex)main.do 를 그대로 쓰지 않게

왜냐하면 어느 부분이 바뀌어야 하는지 고려 했을 때 5분 안으로 처리해야 하기 때문에

빠르게 할수록 잘하는 것.

나스로 다 올릴 것

-- 주의해야 할 사항 --

- 초급들이 많이 하는 모습

: 커밋은 퇴근시간에 하면 개발서버에 올라가는데 3시간정도 발생하게 된다. 이때 베포툴이 3시간마다 돌면서 쌓아놨던걸 개발서버에 올리게 됨, 익스플로러도 테스트를 해야 오류가 발생하지 않는데 .. 결과적으로 개발 서버까지 꼭 확인하고 진행 해야 함

- 프로젝트를 새로 만들거나 받아올때는 그냥 돌리는게 아니라 클린을 해야한다

클린 이전에 패키지 익스플로어에 가서 먼저 확인하고 진행할 것

- 내 방에 있는 프로젝트를 체크아웃받고 싶으면 프로젝트를 체크아웃하는게 아니라 svn://를 체크하게되면 모든 프로젝트를 가져오게 되고, 이때 모르는 상태로 서버를 올리려고하면 화면표시 안되니 이런 부분 주의할 것

- 프로젝트를 받고서 내 나스방과의 연결을 끊었던 것을 다시 연결하고 싶으면 원래 연결했던 방법인 팀>쉐어 프로젝트 로 하려고하면 누를수는 있으나, 그렇게 하면 수정되고 있던 프로젝트가 날아가고 현재 올릴라고 했던게 올라감 따라서 수정한 부분 메모장에 넣고 새로 받아서 수정해야함

- 연습을 많이하면 프로젝트가 많이 쌓이게 되니까

폴더를 만들고 폴더별로 정리한다.

svn의 내방에 > 뉴 > 폴더 // 로 정리 (( ! 내가 정리해서 올릴 것 ))