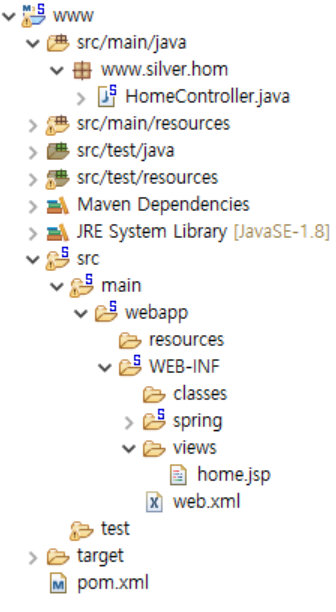
1. spring legacy디렉토리 구조(maven build )
   1. 자바 소스 위치
   2. 정적 컨텐츠 위치
   3. view 파일 위치
   4. 설정파일 위치
   5. Maven 설정파일
   6. Mybatis 설정파일 위치

서비스 정의 :메인화면 보여주세요//404는 처리불가하다는 뜻. 500번은 처리할수 잇는데 문제있음

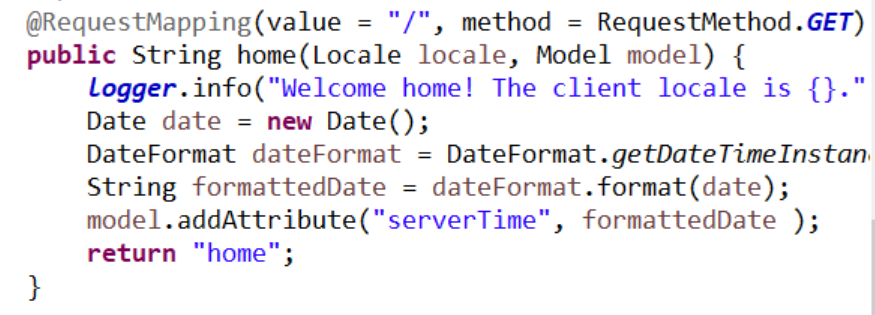
클라이언트 정보

url : /

method : get

parameter : 없음

매칭시켜준다

서버 처리

mapping : /

데이터베이스 처리 없음

view : home

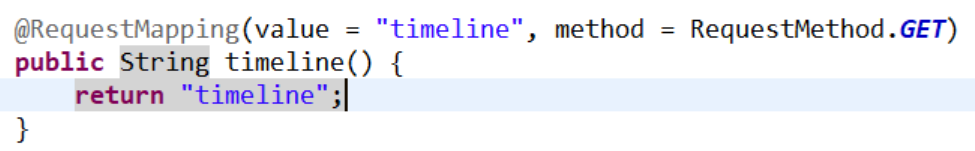
확인문제

서비스 정의 : 시간표 보여 주세요

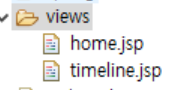
클라이언트 정보 : url : /timeline method: get parameter : 없음 > http://localhost:80808/hom/timeline

서버처리 : 데이터베이스 작업없음.. view : timeline

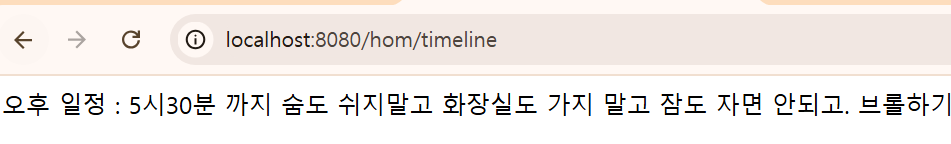
문제> 서버에서 코딩하시오



view가 없어서 만들어 줘야 한다.



서버를 재시작하여 테스팅 한다.



서비스 정의 : 지각자만보기 기능.

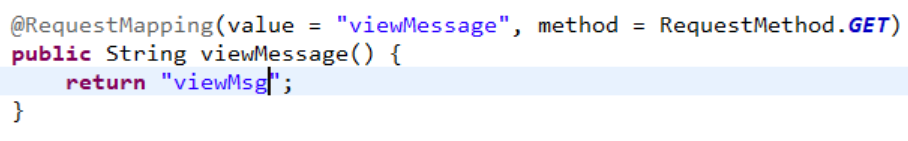
시간표 보여 주세요 서비스에서 <지각자만보기>라는 메뉴가 있다.

지각자만보기 라는 메뉴를 클릭하면 내일은 일찍오세요 라고 응답한다.

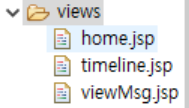
클라이언트정의 : url: /viewMessage metho: get parameter :없음

URL : http://서버주소:port/hom/vewMessage

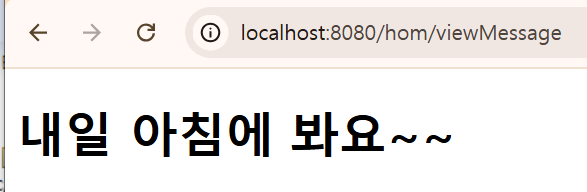
서버정의: 데이터베이스 작업없음.. view : viewMsg



뷰가 없으니 만들자.



테스팅해보자



시간표 보여 주세요 서비스에서 <지각자만보기>라는 메뉴가 있다 이렇게 정의해서

이대로 코드를 수정해 볼께요..

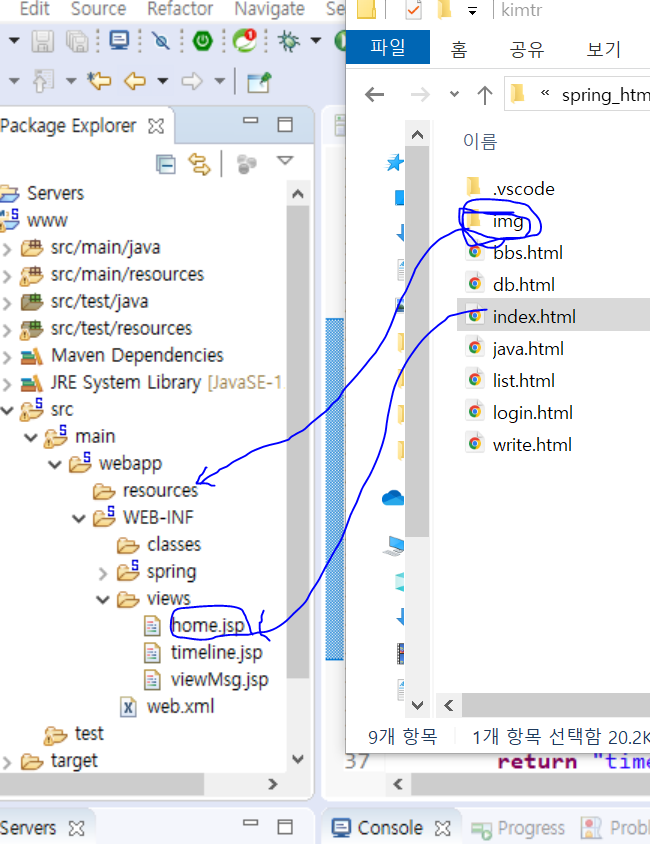
해결) 시간표 view에서 지각자만보기 메뉴를 추가 시키겠다.



테스팅은 timeline 요청부터 시작합니다.

a 태그의 request method는 get방식입니다. 주소창에 직접 입력하셔도 get방식입니다.

이번에는 샘플로 제공한 html 파일을 스프링에 이식한다.



11월 07일 작업은

클라이언트에서 url, method, parameter를 전달하고 서버는 처리를 하는 과정입니다.

포인트는 1.파라미터를 전달하는 방식인 get, post방식을 이해하고

코드로 작성 할 수 있다.

2. 서버는 파라미터를 어떻게 받아(저장한다)내는지 그 방법을 이해하고

코드로 작성할 수 있다.

첫번재, 클라이언트에서 파라미터를 get 방식으로 전달하는 법

<a href=”/login?uid=kkk&pw=1234”>

<form action=”login” method=”get”>

<input type=”text” name=”uid”>

<input type=”password” name=”pw”>

</form>

두번째, 클라이언트에서 파라미터를 post방식으로 전달하는법

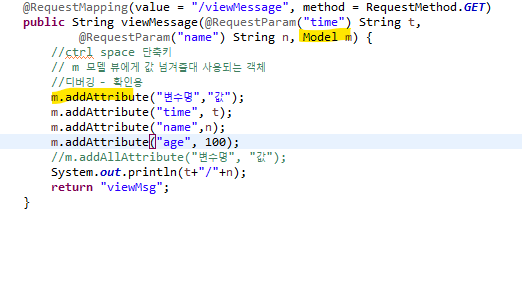
<form action=”login” method=”post”>

<input type=”text” name=”uid”>

<input type=”password” name=”pw”>

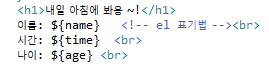
</form>

이제는 컨트롤러가 뷰에게 데이터를 전달하는 예제를 살펴보자



뷰에서는 전달 받은 데이터를 출력합니다. 사용하는 문법은 el표기법, jstl문법

jsp template는 el표기법, jstl문법을 사용할 수 있다.



Spring 구조

-filter dipatcher 등

Spring 주요용어

DI

컨테이너

aop

pojo

11월08일

오늘 수업은

mybatis 퍼시스턴스 프레임워크를 사용하여

spring프레임 워크와 연동을 한다.

퍼시스턴스 프레임워크는 영속적인 의미를 포함하고 있으며

이 의미는 데이터베이스 작업의 의미로 이해하면 됩니다.

즉, 데이터베이스 작업을 원활히 할 수 있는 프레임워크를

퍼시스턴스 프레임 워크라고 이해하시면 됩니다.

프로젝트 마다 다르지만

어떤 프로젝트는 mybatis를 어떤 프로젝트는 jpa를 사용하기도 합니다.

즉 프로젝트마다 다를 수 있다.

그러니.. 퍼시스턴스 프레임 워크의 성격을 이해하는 것은

업무의 흐름을 이해하는데 도움이 됩니다.

수업을 듣고 난 후 셀프 성취도 체크 기준.

수준3: 과정을 이해한다.

수준2: 과정을 이해하고, spring 프로젝트의 구조와 연계하여 설명이 가능하다.

수준1: 과정이해, 구조연계이해, 특징까지 이해하고 문서나 말로서 설명가능.

특: 결합도를 낮춘다는 것의 의미 이해

Mabatis 작업 과정

1. Maven 에서 Mybatis를 사용하기 위해 필요한 라이브러리 설정.

pom.xml 에서 설정하면 자동으로 라이브러리를 다운로드 받아 준다.

2. 컨테이너에게 객체를 등록하기 위한 환경설정을 한다.

3. 서비스레이어와 dao레이어를 만든다.

4. 데이터베이스 작업을 위한 쿼리문을 정의한다. 이것을 mapper 설정이라고 함.

\*\* 순서가 절대적이지 않다는 것을 미리 공지합니다. \*\*;

mybatis 퍼시스턴스 프레임워크

1) 의존성 설정 <- dependency 추가(library 빌드)

maven - pom.xml

1)spring version 확인

----------------- 버전에 따라 어노테이션 먹히는게 다름

java version 확인

dependency 종류

2) 필요한 bean설정(root-context, servlet-context)

bean은 객체

a)datasource b)factory c) template --->추가; 콘솔에 쿼리문출력하는 bean이있다.

3) client ㅡ> 컨트롤러 ㅡ>서브 ㅡ>dao

spring

컨테이너

datasource

~~~~

~~~

실행될때 spring 안 컨테이너 에서 자동으로 객체 생성함

그 생성된 객체의 주소를

inject autowired 등으로 주입 받아서 사용가능.

객체

component bean @어노테이션으로 컨테이너에 객체 생성 가능

11월11일

1. 클라이언트가 요청한 value

2. 404가 뜨는이유

실습을 하면서 정리하자.

1) value = board

2) 컨트롤러가 value을 인식할수 없음

즉, 큰트롤러에 정의가 되어 있지 않다.

이 가능성이 제일 높다.

코드를 작성하다가 404를 만나면

컨트롤러를 체크 하면 된다.

5-1

root-context.xml

생성시점: 앱시작시 contextloaderleader에 의해 생성

역할: 설정파일임

범위 : 앱 전역

datasource

sqlsessionfactory

sqlsessiontemplate

servelt-context.xml

생성시점: 가 DispatcherServlet에 의해 생성

역할: 웹요청과 관련된 설정 관리(컨트롤러 , 뷰 리졸버등)

범위: 각 dipatcher에 국한

11월12일

클라이언트

오늘 최종 정리해 보세요

1. 스프링 레이어 의미

웹 레이어, 서비스 레이어, 리포지토리(dao) 레이어

2. 게시글 쓰기를 클라이언트, 컨트롤러, 서비스, 리포지토리, mapper db

다시 db 리포지토리, 서비스, 컨트롤러, 뷰 순으로 정리.

3. 게시글 전체보기를 클라이언트, 컨트롤러, 서비스, 리포지토리, mappe db

다시 db 리포지토리, 서비스, 컨트롤러, 뷰 순으로 정리

4.삭제 기능을 클라이언트, 컨트롤러, 서비스, 리포지토리, mapper db

다시 db리포지토리, 서비스, 컨트롤러, 뷰 순으로 정리

게시글 쓰기

클라이언트 action: bwrdo , method: post

컨트롤러 postmapping ,@modelatribute bovo bovo

vo객체생성, 클라이언트가 보낸 name들과 같게

서비스

addboard(boaddvo boadvo)throws exception

if(board.getviewmemember) == null)

else

}

boadddo.insertboard

리포지토리

mapper db

@Contoller 어노테이션을 붙이면 핸들러가 스캔할 수 있는 빈(Bean) 객체가 되어 서블릿용 컨테이너에 생성됩니다.  마찬가지로 @Repository, @service 어노테이션은 해당 클래스를 루트 컨테이너에 빈(Bean) 객체로 생성해주는 어노테이션입니다.

참고로 객체 내에서 데이터 변경 작업이  있는 VO(DTO) 객체와 같은 경우는 동기화 문제로 인해 Bean 객체로 사용하지 않습니다. Bean 객체는 항상 데이터 변경이 없는 객체에 한해 사용하는 점에 유의해야 합니다.

컨트롤러 : @Controller  (프레젠테이션 레이어, 웹 요청과 응답을 처리함)

로직 처리   : @Service   (서비스 레이어, 내부에서 자바 로직을 처리함)

외부I/O 처리 : @Repository (퍼시스턴스 레이어, DB나 파일같은 외부 I/O 작업을 처리함)

@컨트롤러는 동기방식 컨트롤러(대개)

그런데

@컨트롤러 앞에 rest컨트롤러 붙은거는

비동기 방식 다 사용함.

1. @컨트롤러 - 동기방식

2. @RestController - 비동기방식