스프링

Loc : 제어권의 역전. 컨테이너 프레임워크가 객체를 관리하는 구조. 모든 제어권은 컨테이너가

Loc 종류

Dependency lookup : 빈을 룩업하여 컨테이너 종속성 증가

>> setter injection, constructor injection

DI : 의존성 주입.

AOP : 관점지향 프로그래밍. 문제를 바라보는 관점을 기준으로 프로그래밍하는 기법. 공통기능으로 핵심관심사항으로 나뉜다.

공통관심사항 : 공통기능으로 전반에 걸쳐 필요한거. 예)로깅, 트랜잭션, 보안

핵심관심사항 : 핵심로직, 핵심 비즈니스 로직. 예) 계좌이체, 이자계산, 대출처리 등

AOP에 쓰이는 용어

1. **Advice 2. Joinpoint 3. Pointcut 4. Advisor**

Advice : 언제 어떠한 기능의 공통 관심로직을 핵심 로직에 적용할지를 정의한다.

Joinpoint : 실행 처리 플로우에서 advice를 위빙하는 포인트. 메서드 호출이나 예외 발생 포인트등…

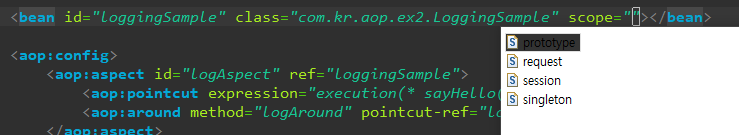
Pointcut : 하나 또는 복수의 joinPoint를 하나로 AR은 것. 하나의 pointcut에는 복수 advice를 연결할 수 있다. 하나의 advice를 복수 joinpoint에 연결도 가능

Advisor : advice, pointcut을 하나로 묶어 다루는 것. Advisor는 스트링에만 있는데 AOP에서 관점을 나타내는 개념

Aspect : 여러 객체에 공통으로 적용되는 공통 관심 사항. 로깅, 트랜잭션 처리들이 있다.

Weaving : advice를 핵심 로직 코드에 적용하는 것.

**Scope의 종류**



**경량컨테이너, 역제어 컨테이너, IoC컨테이너 = 스프링 컨테이너 가 관리한다.**

<**context:component-scan** base-package=*"com.kt.di"* />

이 페이지 밑에있는거는 관리한다.

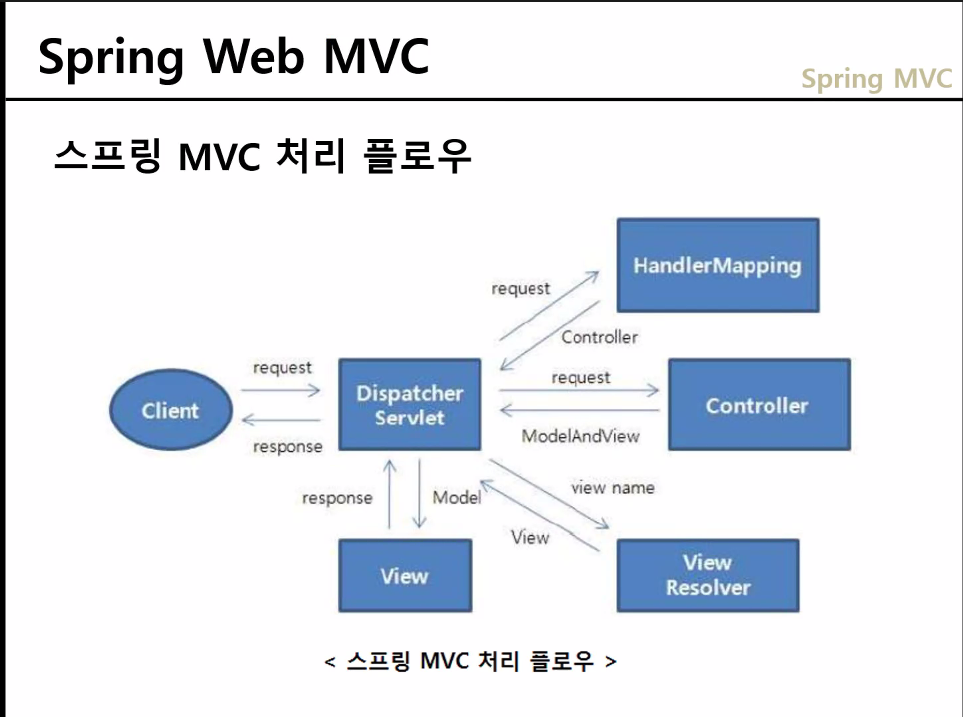
**@컨트롤러, 서비스, 컴포넌트** 어노테이션 >> 자동 생성된다. 이건 반드시 베이스패키지에 만들어야한다. 특정페이지 밑에있는 자동 생성 어노테이션.

**그룹아이디**는 패키지의 두개

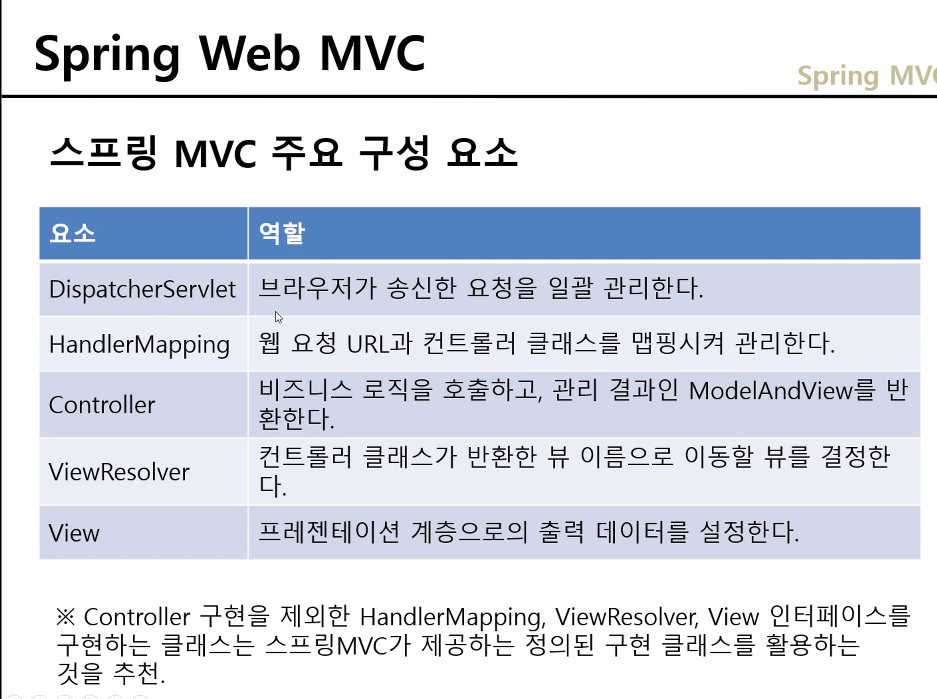
**아디팩트** 는 컨텍스트 루트, 패키지의 뒤에거로 이름 명명

**POJO**

**Aspect는 관심노드**



모든 요청은 컨트롤러가 받는것처럼 보이지만 디스패쳐 서블릿이 받는다. 구조!



노출되어있는거는 뷰리저브

MVC중에 사용자의 요청을 받는게 컨트롤러. 스프링에서 컨트롤러 대신 받는거 맨 앞에서 디스패쳐 서블릿이 받는다.

0131

Pom.xml은 처음 di주입 >> 인터넷에서 다운

Web.xml에 param으로 연결하는거 설정해줌

config파일 밑에 xml파일 설정

sqlsessionfactory : 팩토리빈에

<**property** name=*"typeAliases"*>

<**list**>

<**value**>com.kt.bank.dto.Account</**value**>

주입하면

**insert** id=*"insertMember"* parameterType=*"com.kt.bank.dto.Member"*

파라미터 타입에 Member로만 써도 된다.

시험 : pom.xml은 준다. Maven 업데이트 해야함

DB정보는 주는데 자신의 아이디, 비밀번호 연결해서 써야함

Web.xml : filter로 인코딩 처리. Context-param 이름이 같은이유 >> 같이 연결하겠다 param-name >> listener로 연결

Sql 연결 : sqlsessionfactortbean 에다가 연결 : root-context.xml 로 연결연결된다.

db-properties 맞는 위치로 연결 classpath / db-properties 주는데 이름이 다를수 있다.

**constructor-arg : dao로 쓴다.**

Board.xml에서 dtd 복사

필기시험

DI주입 > 세터로 주입 **property** 에서 파일 설정

**@Required** : **세터함수**에 붙어있으면 Bean객체를 생성할 때 프로퍼티를 넣어주라는 강제성

**contextLoaderListener** : 서블릿의 공통모듈, 라이브러리는 담는 클래스

<콘텍스트 파람에 들어가는 파람 네임음 contextConfigLocation이고 밑에param name이랑 같아야한다.

Filter는 인코딩과 같이 해야되는데 encoding Filter 이름은 CharacterEncodingFilter

@Controeller, Service, Component, Repository : 자동으로 스프링에서 생성된다.

>> servlet-root에 context component 루트 밑에 다 있어야 한다.(컴포넌트 스캔!!)

<**context:component-scan** base-package=*"com.kt.board"* />

모든 요청은 **DispatchServlet이** 받아서 처리한다. 그러나 컨트롤러는 아님

**POJO**

**Scope의 의미 : portotype – 매번 필요할 때 생성 / request-이전 다음페이지 공유/session-특정 사용자/singleton-하나만**

**스프링 컨테이너 : 역제어 컨테이너 / IOC컨테이너 : 빈객체의 생성 소멸과 관련된거를 관리함**

