

1) 스프링 빈, ApplicationContext를 정의하시오.

1) - ① 스프링 빈

스프링 컨테이너가 관리하는 객체이다. 의존 관계의 그 대상이라고 말할 수 있다. 스프링 빈은 애플리케이션의 핵심을 이루는 객체이며, Spring IoC(Inversion of Control) 컨테이너에 의해 인스턴스화, 관리, 생성된다. Beans는 우리가 컨테이너에 공급하는 설정 메타 데이터(XML 파일)에 의해 생성된다.

1) - ② ApplicationContext

스프링 코어 모듈안에 구현체들이 많이 들어 있는데 어플리케이션 개발자 입장에서 보면 그 안에 들어가 있는 인터페이스로 BeanFactory, ApplitionContext가 있다. ApplicationContext는 BeanFactory의 확장이라고 보면 된다. BeanFactory 인터페이스를 상속받은 하위 인터페이스로 확장된 기능을 제공한다(internationalization, AOP, Transaction Management). XML 파일로부터 설정 정보를 활용하는 가장 많이 사용되는 클래스이다.

2) WebApplicationContext가 생성되는 2가지 방식을 설명하시오.

웹 애플리케이션을 위한 ApplicationContext이다. ApplicationContext와 WebApplicationContext는 만드는 방식이 다르다. ApplicationContext는 자바의 main을 실행 즉 개발자가 시작을 할 수 있지만, WebApplicationContext는 개발자가 시작을 하지 못하고 톰캣이기 때문에 톰캣이 시작하면 그 때 이벤트를 받아 결과를 만들게 된다. WebApplicationContext이 생성되는 방식은 2가지가 있다. 밑의 2가지는 스프링 프레임 워크에서 제공이 된다.

2) - ① ContextLoaderListener에 의해 생성되는 WAC

스프링 빈 객체와 그것들의 의존관계를 만든다. Persistence(DAO), Service관련 스프링 빈들을 등록하고, 웹 어플리케이션 전체에서 사용할 WAC 객체를 생성한다. root-context.xml에서 설정하면 된다.

2) - ② DispatcherServlet에 의해 생성되는 WAC

ContextLoaderListener에 의해 WAC가 생성되면 그것을 상속받아 DispatcherServlet이 WAC를 만든다. 컨트롤러와 같은 서블릿 관련 빈을 등록하고 해당 서블릿 마다 사용할 WAC 객체를 생성한다. 전체적인 객체와 의존관계를 쥐고 사용한다고 생각하면 된다. 사용자가 요청을 하게 되면 이 DispatcherServlet의 WAC가 먼저 받게 된다. servlet-context.xml 파일에서 설정을 한다.

3) DI ANNOTATION 설정 방법에서 <context:annotation-config />와 <context:component-scan base-package="명"/> 차이점을 설명하시오.

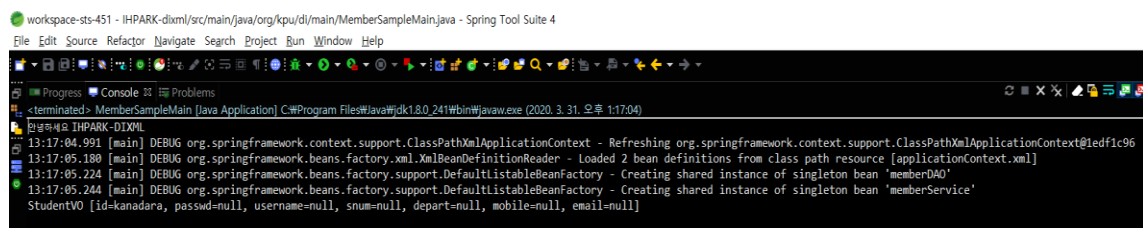
3) - ① <context:annotation-config />

@Autowired 설정을 사용하기 위해서 필요한 설정이다. <context:component-scan />가 설정되어 있으면 생략 가능하다. @Autowired는 컨테이너가 빈과 다른 빈과의 의존성을 자동으로 연결하도록 하는 수단이다. 인스턴스 변수(빈) 앞에 @Autowired 애노테이션을 붙이면 해당 타입의 Component(빈)를 찾아 그 인스턴스(빈)를 주입시켜준다.

2) - ② <context:component-scan base-package="명" />

@Component 애노테이션이 붙은 클래스를 자동으로 빈으로 등록하는 기능이다. @Component는 컨테이너가 인젝션을 위한 인스턴스(빈)을 설정하는 수단이다. 클래스 선언 앞에 @Component를 붙이면 컨테이너가 찾아서 관리하고 @Autowired가 붙은 인스턴스 변수(빈)에 주입시켜준다.

4) di-xml.zip을 참고하여 실습하고 결과 출력 페이지를 PrintScreen하여 첨부하시오.



```
workspace-sts-451 - IHPARK-dixml/src/main/java/org/kpu/di/main/MemberSampleMain.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Progress Console Problems
terminated> MemberSampleMain [Java Application] C:\Program Files\Java\jdk1.8.0_241\bin\javaw.exe (2020. 3. 31. 오후 1:17:04)
원장동생의 IHPARK-DIXML
13:17:04.991 [main] DEBUG org.springframework.context.support.ClassPathXmlApplicationContext - Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1edf1c96
13:17:05.180 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loaded 2 bean definitions from class path resource [applicationContext.xml]
13:17:05.224 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'memberDAO'
13:17:05.244 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'memberService'
StudentVO [id=kanadara, passwd=null, username=null, snum=null, depart=null, mobile=null, email=null]
```