3번째 웹서버?

Di 자동설정과 빈 생명주기와 범위

자동설정

1. @Autowired 어노테이션 있는 객체에 빈 자동으로 셋팅됨 객체 입력 설정 없이도

아래 Namespaces 탭에 context체크후 가능

* Xml 파일 맨위 <context:annotation-config/> : 파싱할떄 어노테이션 해석하라는 명령어

스프링 빈 생명주기

interface이용한 생성주기 실행 메소드

해당 클래스의 자바 파일에서 implements InitializingBean, DisposableBean, EnvironmentAware

InitializingBean 빈 객체 생성하자마자

DisposableBean 빈 객체 소멸 직전

EnvironmentAware 빈 객체 생성

어노테이션 이용한 생성 주기 실행 메소드,

@PostConstruct 빈 객체 생성하자마자

@PreDestroy 빈 객체 소멸 직전

빈 객체에 스코프범위 설정해서 객체생성 가능

bean 객체 생성시 스코프범위 설정할수있는데 singltone & prototype

singltone은 생성된 객체의 주소를 가져와서 하나의 객체만 공유하는데

prototype은 생성된 객체의 정보를 복사해서 가져옴

그래서 prototype형의 객체를 여러 개 가져와 정보를 수정하면 각기 다른 객체 정보로 저장됨

Singltone형은 수정하면 해당 객체의 주소로 생성되기떄문에 수정된 정보를 공유함

1. xml파일에서 빈생성으로 di
2. bean태그 생성시 아래 namespaces ‘c’ ‘p’

c = 생성자 함수 매개면수 p = 필드

AbstractApplicationContext ctx = new GenericXmlApplicationContext(asd, zxc);

컨테이너 만듬/ 환경 셋팅함 속성 1…..

implements

InitializingBean 생성시

DisposableBean 소멸시

setEnvironment 생성시 먼저

Db.properties db정보

env객체를 가지고 오기위한 컨테이너

컨테이너 만들면 속성변수 생김

속성과 짝을 이룬 속성 변수가 있음 ??

Properties란 envl가 읽을수 있는 정보들 모인 파일 확장자?

<!-- 확장자가 properties파일을 가져올수 있는 키워드 -->

<context:property-placeholder location="classpath:META-INF/ex2/admin.properties,

classpath:META-INF/ex2/sub\_admin.properties"/>

profiles 이란

외부 파일을 이용한 설정

Environment 객체를 이용한 스프링 빈 설정

Environment객체안에는 여러 가지 정보들(환경설정을 위한)이 있다. 이런 값들을 관리하기 위해 이런 정보들이 바로 들어있지 않고, PropertySource라고 하는 객체가 있고, PropertySource안에 정보가 있다. Environment 객체로부터 env.getPropertySources()를 하여 데이터를 관리해 주는 PropertySources 객체를 얻는다.

프로퍼티 파일을 이용한 설정 ;

Environment 객체를 사용하지 않고 프로퍼티 파일을 직접 이용하여 스프링 빈을 설정하는 방법 - 스프링 설정 XML 파일에 프로퍼티 파일을 명시하는 방법

프로파일(profile) 속성을 이용한 설정 ;

동일한 스프링 빈을 여러 개 만들어 놓고 상황(환경)에 따라서 적절한 스프링 빈을 사용할 수 있게 합니다. profile 속성을 사용하면 됩니다.

Ex) 개발중 환경 : 127.0.0.1:8090 / 운영중 환경 : 168.165.10.30:80

환경다르게 할 때 xml두개? 빈객체 이름은 같게

처음 빈태그 안에 ex) profile="run"> profile다르게 구분

//Autowired 범위 필드도 적용? 자료형을 보고 달라붙음

== 필드에는 적용되지않고 패키지명을 보고 가장 위에있는 객체를 불러옴

// 프로토타입형식으로 생성된 객체는 어디에 저장 == 메모리 어딘가

//GenericXmlApplicationContext ctx = new GenericXmlApplicationContext() 해당 Profiles를 활성화 언제 xml실행시 ?

//매개변수로 넎으면 바로 파싱?

// AbstractApplicationContext 여러 개 생성해서 하나씩 xml갖는 거랑

하나의 AbstractApplicationContext에 여러 xml갖는것

// placeholder 만 el씀 ==el이 아니라 placeholder만 쓰는 문법

//어노테이션 검색 태크 <context:annotation-config/>

모든 종류 어노테이션 == 객체 생성을 위한 어노테이션만 검색

컨테이너 생명주기

aop

공통기능수행?? 부가적으로 하려는 기능??

핵심기능 수행방법 두개

어노테이션

xml