

Week1. 스프링 핵심원리 기본편 스터디 OT

우리는 무엇을 공부할것인가?



2021. 06. 26(Sat)

Covenant Ko



inflearn

로켓스터디 저장소

Rule

2회 결석시 Out 제도(지속하기 어려운 경우 빠르게)

Rule

2회 결석시 Out 제도(지속하기 어려운 경우 빠르게)

서로의 실력과 학습력이 다름을 이해

Rule

2회 결석시 Out 제도(지속하기 어려운 경우 빠르게)

서로의 실력과 학습력이 다름을 이해

부담없이, 자신이 공부한 것을 이야기하고 피드백 받는 시간

Rule

2회 결석시 Out 제도(지속하기 어려운 경우 빠르게)

서로의 실력과 학습력이 다름을 이해

부담없이, 자신이 공부한 것을 이야기하고 피드백 받는 시간

모르는 것은 부끄러운 것이 아니다!!

함께 스터디를 하자!

함께 스터디를 하자!

소개의 시간

함께 스터디를 하자!

함께 **스터디**를 하자!

함께 **스터디**를 하자!

學習

무엇을 공부할까? 무엇에 투자해야하나?



무엇을 공부할까? 무엇에 투자해야하나?



빈 스코프

빈스코프의 종류

1. 싱글톤 스코프

- 스프링빈은 기본적으로 싱글톤스코프로 생성되는데 이는 스프링 컨테이너의 시작과 종료될때까지 유지된다.

2. 프로토타입 스코프

- 빈의 생성과 의존관계까지만 관리해주고 그이후는 관리하지않는 짧은 스코프

프로토타입 스코프의 빈은 같은 요청이라도 요청을 받을때마다 새로운객체를 생성한다.

초기화 까지만 관여하기 때문에 @PreDestory 같은 종료메서드는 호출되지 않는다.

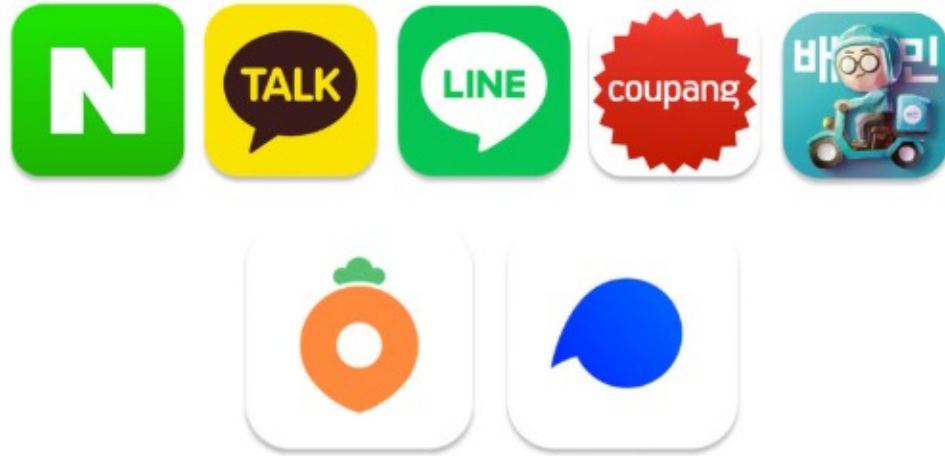
즉 종료메서드에 해당하는 작업은 클라이언트가 직접 처리하도록한다.

★프로토타입 스코프와 싱글톤 빈과 함께 사용하는 경우 문제가 발생한다.

정확히는 싱글톤빈이 프로토타입빈을 사용하면 문제가 생긴다. 생성시점에만 의존관계 주입을 받기 때문에, 새로 생성된 프로토타입빈이 싱글톤빈과 함께 계속해서 유지되고 있는상태가 된다.

무엇을 공부할까? 무엇에 투자해야하나?

→ 개발자는 무엇을 알아야하나? 목적! 뭘하는데?



SUMMARY

@SpringBootApplication 애너테이션은 @ComponentScan 애너테이션과 @EnableAutoConfiguration 애너테이션을 포함하고 있다. @ComponentScan 애너테이션은 프로젝트 패키지에 있는 Bean을 찾아서 등록하고, @EnableAutoConfiguration 애너테이션은 의존성 라이브러리를 기반으로 사용 가능성이 높은 Bean을 추측해 자동으로 등록한다.

그다음으로 RTS-API 프로젝트의 Java 소스 폴더에 WebMvcConfigurerAdapter 클래스를 상속(extends)한 MvcConfig 클래스를 추가한다. @Configuration 애너테이션으로 WebMvcConfigurerAdapter Bean을 생성하면 기본적인 설정이 모두 자동으로 이루어지기 때문에 **web.xml** 파일의 ContextLoaderListener 설정과 DispatcherServlet 설정, CharacterEncodingFilter 설정은 Spring Boot에서는 추가하지 않아도 된다.

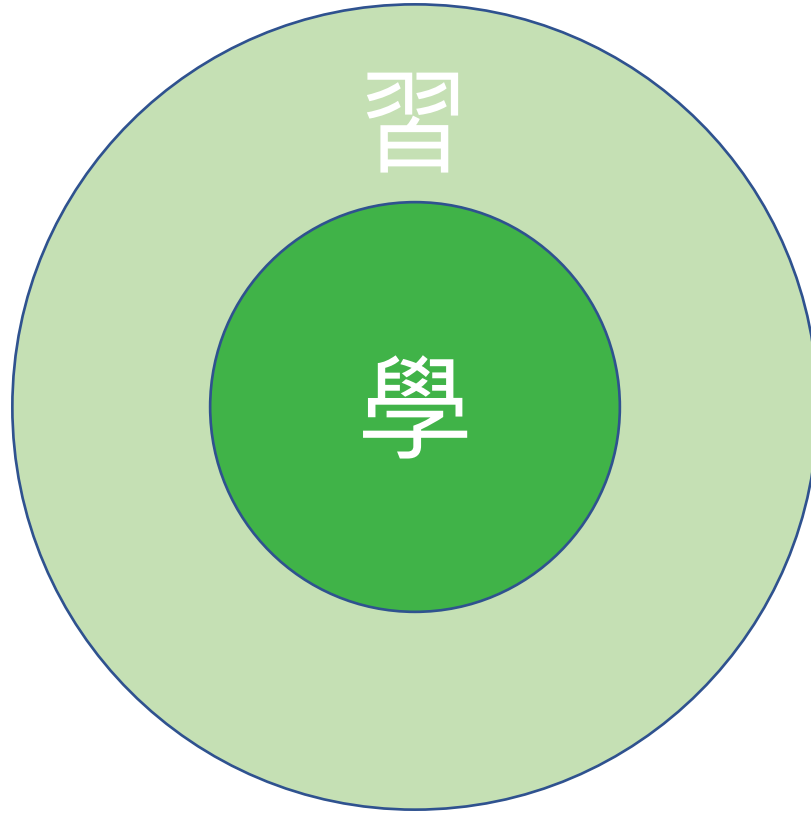
```
@Configuration
public class MvcConfig extends WebMvcConfigurerAdapter {
}
```

Spring Boot에서는 Bean으로 생성된 서블릿과 필터를 Spring 컨테이너에 자동으로 등록한다. 따라서 **web.xml** 파일에 있던 customFilter 등록 관련 설정을 삭제할 수 있다.

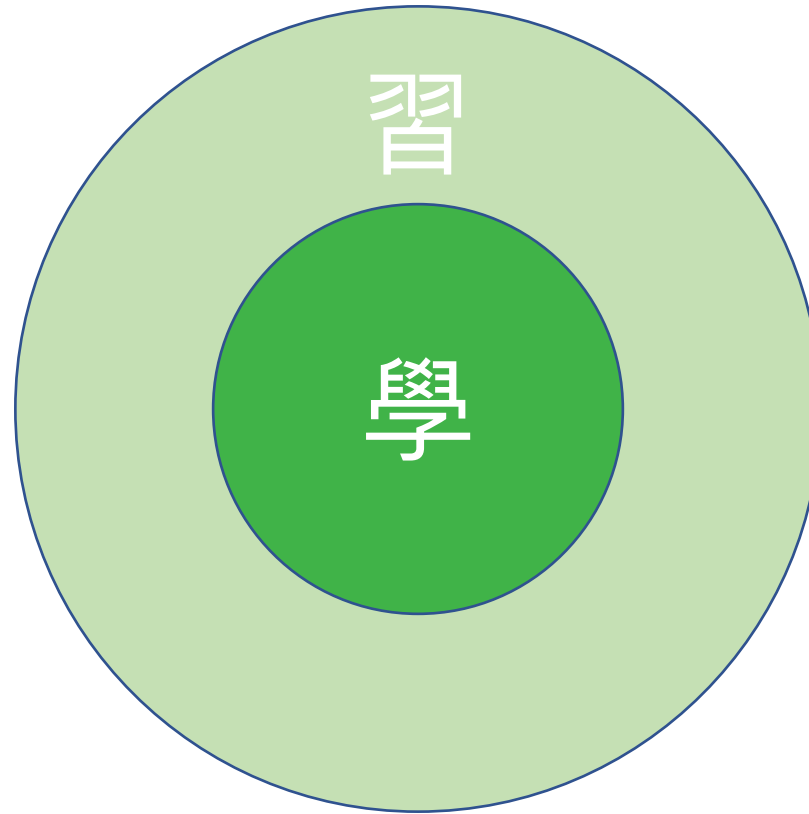
네이버 동영상 플랫폼 개발 프레임워크의 Spring Boot 전환기

<https://d2.naver.com/helloworld/5626759>

무엇을 공부할까? 무엇에 투자해야하나?



무엇을 공부할까? 무엇에 투자해야하나?



실무에서 어떻게 쓰이는가?
나는 왜 사이드 프로젝트할 때 ~을 사용했는가?
사용법을 넘어 장단점을!

취업 준비의 시간

=

좋은 개발자가 되기 위한 시간

왜 스터디를 하시나요?