

IOC Container

Sechung

IOC Container?

- Inversion of control(제어의 역전)을 구현한 컴포넌트 또는 프레임워크
 - 인스턴스 관리를 개발자에서 컨테이너로 역전
- How?
 - Dependency Injection
 - Service Locator Pattern

Dependency Injection

- 객체가 그들의 의존성을 외부에서 주입받는 것을 의미
- 자바에서 객체 외부에서 의존성을 넣으려면
 - 생성자 인수
 - 팩토리 메서드 인수
 - 팩토리 메서드 반환 인스턴스

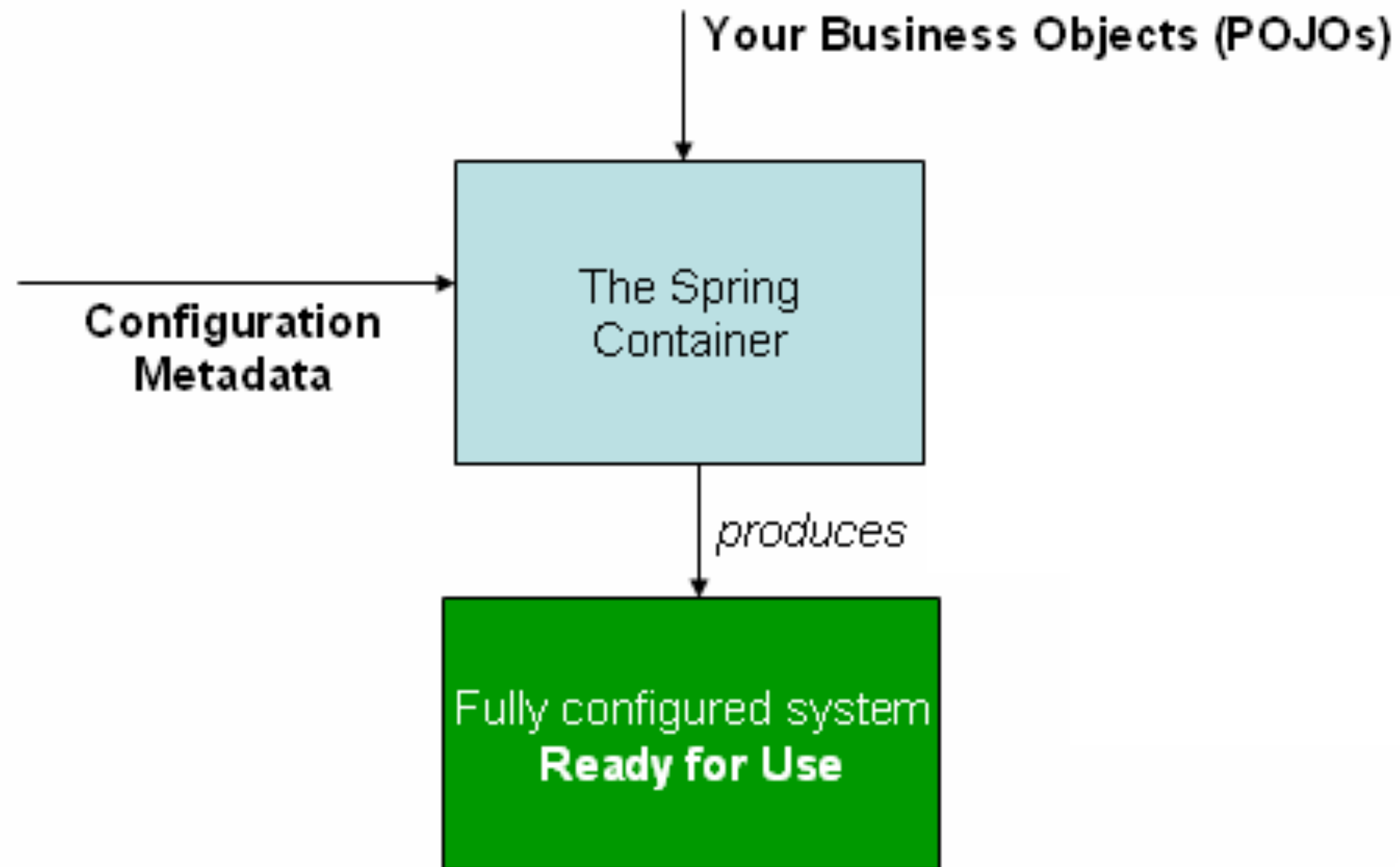
Spring Dependency Injection

- 생성자 주입
 - final 키워드를 활용하여 불변성 보장
- setter 주입
 - 객체 생성 이후 의존성 변경 가능
- 필드 주입
 - @Autowired
 - 강한 컨테이너 의존성

Service Locator Pattern

- 의존성을 관리해주는 별도의 객체를 만드는 패턴
- 해당 객체를 Service Locator라고 한다

IOC의 작동



- 순수 자바 객체 POJO
- 관리 정보를 담은 메타데이터

Configure metadata

- XML
- JAVA Code
- Annotation

XML metadata

```
<beans>
  <import resource="services.xml"/>
  <import resource="resources/messageSource.xml"/>
  <import resource="/resources/themeSource.xml"/>

  <bean id="bean1" class="..."/>
  <bean id="bean2" class="..."/>
</beans>
```

- XML 파일에 관리를 원하는 객체를 작성하여 등록

Java Code metadata

```
@Configuration
public class ServiceConfiguration {
    @Bean(name = "memberService")
    public MemberService memberService() {
        return new MemberServiceImpl();
    }

    @Bean(name = "member42Service")
    public MemberService member42Service() {
        return new Member42Service();
    }
}
```

- @Configuration
- @Bean으로 등록할 인스턴스를 생성

Annotation metadata

- @Component, @Controller, @Service, @Repository, @Configuration ..
 - 클래스 어노테이션을 통해 등록할 객체 명시
- @ComponentScan
 - 스프링 부트가 하위 패키지 스캔 / 혹은 선언된 패키지 스캔

IOC Container?

- Inversion of control(제어의 역전)을 구현한 컴포넌트 또는 프레임워크
 - 인스턴스 관리를 개발자에서 컨테이너로 역전
- How?
 - Dependency Injection
 - Service Locator Pattern

Finish