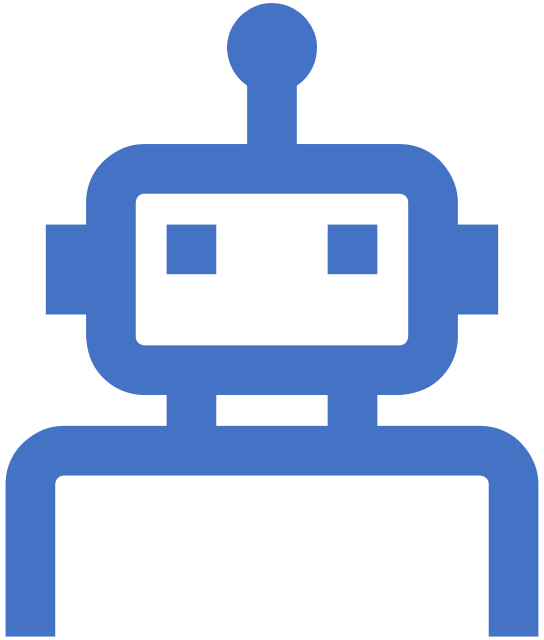


[6조]
2주차 선행학습

[발표자] 이주연

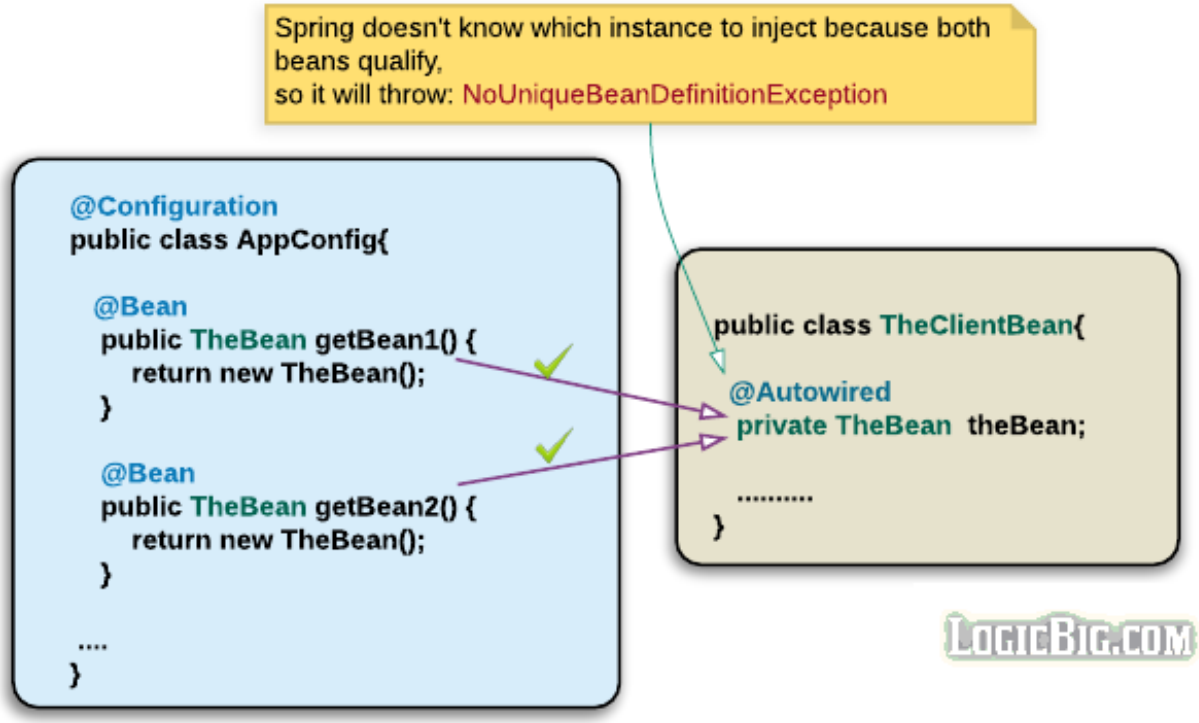
정지운

김현중

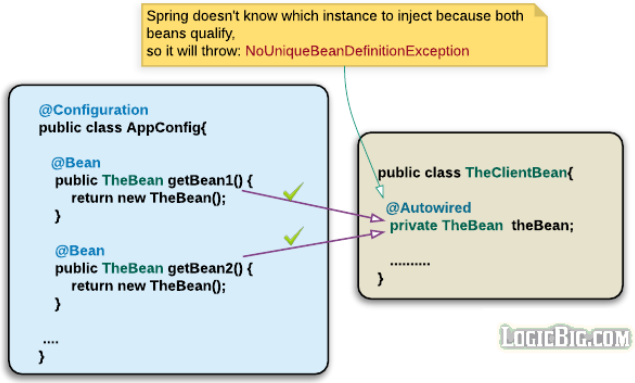
이민형

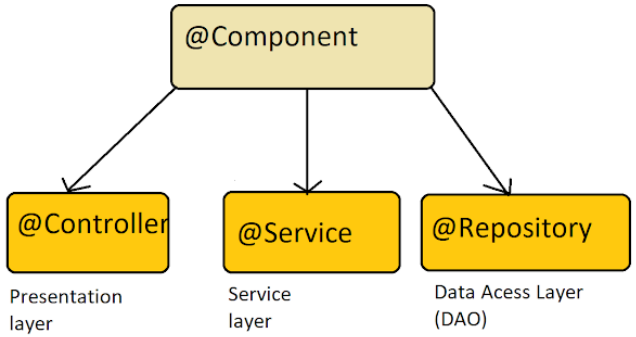
Bean 주입

- Spring Bean은 Spring에 의해 생성되고 관리되는 객체.
- Spring은 Bean을 생성 및 관리를 위해 스프링 컨테이너를 제공
- Spring은 Bean 간의 의존성을 자동으로 주입.
- Spring은 Bean의 생명주기를 관리.



Bean과 Component 차이





@Bean	@Component
개발자가 컨트롤이 불가능한 외부 라이브러리들을 Bean으로 등록하고 싶은 경우 사용된다.	개발자가 직접 컨트롤이 가능한 클래스들의 경우에 사용된다.
메소드 또는 어노테이션 단위에 붙일 수 있다.	클래스 또는 인터페이스 단위에 붙일 수 있다.
@Configuration 어노테이션이 필요하다.	@Configuration 어노테이션이 필요 없다.

Field Injection과 Constructor Injection 차이

- **Field Injection**

```
@Component
public class SomeService {
    @Autowired private SomeOtherService someOtherService;
}
```

  - 클래스의 필드로 의존성을 주입하는 방식
  - 주로 @Autowired와 같은 어노테이션을 필드 위에 선언
  - 코드가 간결
  - 단일 책임의 원칙 위반 등 여러 문제를 야기하여 권장하지 않는 방식

Field Injection과 Constructor Injection 차이

- **Constructor Injection**

```
@Component
public class SomeService {
    private final SomeOtherService someOtherService;

    @Autowired
    public SomeService(SomeOtherService someOtherService){
        this.someOtherService = someOtherService;
    }
}
```

- 의존성 주입을 클래스 생성 시점에 수행하기 때문에 불변성을 유지
- 객체 초기화 시점이 명확
- 의존성이 명확하여 클래스 의존성 관리 용이

@Primary와 @Qualifier annotation

- @Primary

여러 Bean이 존재할 때, @Primary 어노테이션이 붙은 Bean이 우선순위를 갖는다.

- @Qualifier

Bean의 추가 구분자를 붙여주는 방식

같은 타입의 Bean이 여러 개 있으면 추가 구분자를 통해 Bean을 주입

- @Primary와 @Qualifier의 우선 순위

@Qualifier가 @Primary보다 높은 우선 순위를 가진다.