

# Practice #2-2

## Mybatis-Spring 내용보충

Spring이 참조하는 @Configuration이 붙은 설정파일에서 아래와 같이 지정이 되었다고 했을 때,

```
1 package com.el.test.config;
2
3 import javax.sql.DataSource;
4
5 import org.apache.ibatis.session.SqlSessionFactory;
6 import org.mybatis.spring.SqlSessionFactoryBean;
7 import org.springframework.context.annotation.Bean;
8 import org.springframework.context.annotation.Configuration;
9
10 @Configuration
11 public class MybatisConfig {
12     @Bean
13     public SqlSessionFactory sqlSessionFactory(DataSource dataSource) throws Exception {
14         SqlSessionFactoryBean factoryBean = new SqlSessionFactoryBean();
15         factoryBean.setDataSource(dataSource);
16         return factoryBean.getObject();
17     }
18 }
19
```

객체를 만들지도 않았는데 파라미터로 뜬금없이 DataSource라는 클래스의 객체를 넣어줬다.  
(물론 클래스가 javax.sql의 라이브러리에 존재하긴 하지만 아직 이 시점에서는 객체를 만들어준 적이 없다.)

이런 경우 방금 위의 스크린샷에서 SqlSessionFactory 객체를 생성한 것과 같은 방식으로 DataSource를 생성해서 리턴하는 메소드를 만들고 @Bean을 붙여주면 스프링이 알아서 DataSource 객체를 먼저 만들고 SqlSessionFactory의 파라미터로 집어넣어주어 세팅해준다.