

2024년 상반기 K-디지털 트레이닝

Spring Context

[KB] IT's Your Life

✓ 객체의 조립

- 스프링 프레임워크의 설정을 통해 객체 구성
 - 의존성 주입(Dependency Injection)

✓ 프로젝트 준비

- 탐색기에서 기존 ScoulaTodo 프로젝트를 ScoulaTodoSpring으로 복사

settings.gradle (프로젝트명 변경)

```
rootProject.name = 'ScoulaTodoSpring'  
include ':ScoulaCli'  
project(':ScoulaCli').projectDir=new File('C:\\KB_Fullstack\\04_Java\\ScoulaLib\\scoulacli')
```

build.gradle (스프링 추가)

```
...  
dependencies {  
    implementation project(':ScoulaCli')  
  
    implementation 'org.springframework:spring-context:5.3.37'  
  
    ...  
}
```

→ Gradle 동기화

✓ 스프링의 어노테이션

○ @Configuration

- 클래스 레벨
- 지정한 클래스가 스프링 설정 클래스임을 나타냄

○ @Component

- 클래스 레벨
- 지정한 클래스가 스프링 컨텍스트에 등록되어야 함을 나타냄

○ @ComponentScan({"패키지명", ... })

- 클래스 레벨
- @Configuration과 함께 사용
- 지정한 패키지(하위 패키지 포함)에서 @Component 어노테이션을 찾아 컨텍스트에 등록

✓ 스프링의 어노테이션

○ @Bean

- @Configuration 클래스의 메서드 레벨
- 해당 메서드의 리턴 객체를 컨텍스트에 등록
 - 추후 리턴 타입 또는 이름(메서드 명)으로 검색

○ @Autowired

- @Component 클래스의 멤버 필드/메서드 레벨
- 지정한 타입의 객체를 컨텍스트에 찾아 초기화 해 줌
 - 해당 타입의 객체가 컨텍스트에 등록되어 있어야 함

ProjectConfig.java (스프링 설정 파일)

```
package org.scoula.todo.config;

import org.scoula.todo.common.JDBCUtil;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

import java.sql.Connection;
import java.sql.DriverManager;
import java.util.Properties;

@Configuration
@ComponentScan({ "org.scoula.todo" })
public class ProjectConfig {
```

ProjectConfig.java (스프링 설정 파일)

@Bean

```
Connection conn() throws Exception {  
    Properties properties = new Properties();  
    properties.load(JDBCUtil.class.getResourceAsStream("/application.properties"));  
    String driver = properties.getProperty("driver");  
    String url = properties.getProperty("url");  
    String id = properties.getProperty("id");  
    String password = properties.getProperty("password");  
  
    Class.forName(driver);  
    return DriverManager.getConnection(url, id, password);  
}
```


TodoDaoImpl.java

@Component

```
public class TodoDaoImpl implements TodoDao{

    @Autowired
    Connection conn;
    // Connection conn = JDBCUtil.getConnection();
    ...
}
```

UserDaoImpl.java

@Component

```
public class UserDaoImpl implements UserDao {
    @Autowired
    Connection conn;
    // Connection conn = JDBCUtil.getConnection();
    ...
}
```

UserDaoImpl.java

```
@Component
public class LoginService {
    //    UserDao dao = new UserDaoImpl();
    //    UserDao dao = Context.getBean(UserDao.class);
    @Autowired
    UserDao dao;

    ...
}
```

AccountService.java

```
@Component
public class AccountService {
    @Autowired
    UserDao dao;

    ...
}
```

TodoService.java

```
@Component
public class TodoService {
    @Autowired
    TodoDao dao;

    ...
}
```

TodoApp.java (스프링 설정 파일)

@Component

```
public class TodoApp extends App {
```

```
...
```

@Autowired

```
LoginService loginService;
```

@Autowired

```
AccountService accountService;
```

@Autowired

```
TodoService todoService;
```

```
...
```

```
// onClose() 제거
```

```
...
```

ProjectConfig.java (스프링 설정 파일)

```
public static void main(final String[] args) {  
    ApplicationContext ctx = new AnnotationConfigApplicationContext(ProjectConfig.class);  
    TodoApp app = ctx.getBean(TodoApp.class);  
    app.run();  
}  
}
```

✓ 실행 설정

