

2025년 상반기 K-디지털 트레이닝

스프링 컨텍스트- 빈 정의

[KB] IT's Your Life



Spring 프로젝트 틀을 만드세요.

○ 프로젝트명: ex01

o group: org.scoula

o Artifact ld: ex01

build.gradle

implementation 'org.springframework:spring-context:5.3.37'

- 🗸 org.scoula.domain 패키지에 Parrot 클래스를 정의하세요.
 - ㅇ 필드
 - String name
 - o name에 대한 Getter/Setter 추가

domain.Parrot.java

```
package org.scoula.domain;

public class Parrot {
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

- 🗸 org.scoula.config 패키지에 ProjectConfig 클래스를 정의하세요.
 - 스프링의 Configuration 클래스로 정의
 - o Parrot을 빈으로 등록
 - Parrot의 name을 "Koko"로 설정

config.ProjectConfig.java

```
package org.scoula.config;
import org.springframework.context.annotation.Configuration;

@Configuration
public class ProjectConfig {
    @Bean
    Parrot parrot() {
        var p = new Parrot();
        p.setName("Koko");
        return p;
    }
}
```

- 🗸 org.scoula.app 패키지에 Main 클래스를 정의하세요.
 - ProjectConfig를 설정 클래스로하여 Spring Context를 생성
 - 컨텍스트를 이용하여 Parrot 빈을 추출
 - 추출된 Parrot의 name 출력

🗹 app.Main.java

```
package org.scoula.app;
import org.scoula.config.ProjectConfig;
import org.scoula.domain.Parrot;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
public class Main {
    public static void main(String[] args) {
        var context = new AnnotationConfigApplicationContext(ProjectConfig.class);
        Parrot p = context.getBean(Parrot.class);
        System.out.println(p.getName());
```

- 🗸 org.scoula.config 패키지에 ProjectConfig2 클래스를 정의하세요.
 - o 3개의 Parrot 빈을 등록
 - 각각의 메서드와 각가의 Parrot의 name은 다음과 같음
 - parrot1(), "Koko"
 - parrot2(), "Miki"
 - parrot3(), "Riki"
 - o parrot2의 빈 이름을 "miki"로 설정

config.ProjectConfig2.java

```
@Configuration
public class ProjectConfig2 {
    @Bean
    Parrot parrot1() {
       var p = new Parrot();
       p.setName("Koko");
       return p;
    @Bean(name = "miki") // 빈의 이름 등록 @Bean(value="miki"), @Bean("miki")
    Parrot parrot2() {
       var p = new Parrot();
       p.setName("Miki");
        return p;
    @Bean
    Parrot parrot3() {
       var p = new Parrot();
       p.setName("Riki");
        return p;
```

- org.scoula.app 패키지에 Main2 클래스를 정의하세요.
 - ProjectConfig2를 설정 클래스로하여 Spring Context를 생성
 - 컨텍스트를 이용하여 "miki" 이름의 빈을 추출
 - o 추출된 Parrot의 name 출력

Main2.java

```
public class Main2 {
    public static void main(String[] args) {
        var context = new AnnotationConfigApplicationContext(ProjectConfig2.class);

        Parrot p = context.getBean("miki", Parrot.class);
        System.out.println(p.getName());

    }
}
```

Miki

- Parrot 클래스를 컴포넌트 스캔을 통해 빈으로 자동 등록되도록 정의하세요.
 - o @PostConstruct를 이용하여 빈의 name 속성을 "Kiki"로 설정

build.gradle

```
dependencies {
  implementation 'javax.annotation:javax.annotation-api:1.3.2'
  ...
}
```

Parrot.java

```
package org.scoula.domain;
import org.springframework.stereotype.Component;
import javax.annotation.PostConstruct;
@Component
public class Parrot {
    private String name;
    @PostConstruct
    public void init() {
        this.name = "Kiki";
    public String getName() {
        return name;
    public void setName(String name) {
        this.name = name;
```

♡ Parrot 클래스를 컴포넌트 스캔을 통해 빈으로 자동 등록되도록 ProjectConfig3 설정 클래스를 정의하세요.

config.ProjectConfig3.java

```
package org.scoula.config;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
@Configuration
@ComponentScan(basePackages = "org.scoula.domain")
public class ProjectConfig3 {
}
```

- 🗸 org.scoula.app 패키지에 Main3 클래스를 정의하세요.
 - ProjectConfig3을 설정 클래스로하여 Spring Context를 생성
 - o Parrot 빈 추출
 - o 추출된 Parrot의 name 출력

Main3.java

```
package org.scoula.app;
import org.scoula.config.ProjectConfig3;
import org.scoula.domain.Parrot;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
public class Main3 {
    public static void main(String[] args) {
        var context = new AnnotationConfigApplicationContext(ProjectConfig3.class);
        Parrot p = context.getBean(Parrot.class);
        System.out.println(p);
        System.out.println(p.getName());
                                                                          org.scoula.domain.Parrot@d6e7bab
                                                                          null
```



2025년 상반기 K-디지털 트레이닝

스프링 컨텍스트- 빈 작성

[KB] IT's Your Life



🗸 Spring 프로젝트 틀을 만드세요.

○ 프로젝트명: ex02

o group: org.scoula

o Artifact ld: ex02

○ 스프링 코어, annotation-api 의존성 추가

build.gradle

```
plugins {
   id 'java'
group = 'org.scoula'
version = '1.0-SNAPSHOT'
repositories {
   mavenCentral()
dependencies {
    implementation 'org.springframework:spring-context:5.3.37'
    implementation 'javax.annotation:javax.annotation-api:1.3.2'
    testImplementation platform('org.junit:junit-bom:5.10.0')
    testImplementation 'org.junit.jupiter:junit-jupiter'
test {
    useJUnitPlatform()
```

- ◎ org.scoula.beans 패키지에 Parrot 클래스를 정의하세요
 - o String name 속성과 Getter/Setter 추가
 - o toString 추가

beans.Parrot.java

```
package org.scoula.beans;
public class Parrot {
    private String name;
    public String getName() {
        return name;
    public void setName(String name) {
        this.name = name;
    @Override
    public String toString() {
        return "Parrot : " + name;
```

- 🗸 org.scoula.beans 패키지에 Person 클래스를 정의하세요
 - String name 속성과 Getter/Setter 추가
 - o Parrot parrot 속성과 Getter/Setter 추가

beans.Person.java

```
package org.scoula.beans;
public class Person {
 private String name;
 private Parrot parrot;
  public String getName() {
    return name;
  public void setName(String name) {
    this.name = name;
  public Parrot getParrot() {
    return parrot;
  public void setParrot(Parrot parrot) {
    this.parrot = parrot;
```

- 🗸 org.scoula.config 패키지에 ProjectConfig 설정 클래스를 정의하세요.
 - 다음과 같이 빈을 등록
 - Parrot
 - name 속성: Koko
 - Person
 - name 속성: Ella

config.ProjectConfig.java

```
package org.scoula.config;
import org.scoula.beans.Parrot;
import org.scoula.beans.Person;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
@Configuration
public class ProjectConfig {
    @Bean
    public Parrot parrot() {
        Parrot p = new Parrot();
        p.setName("Koko");
        return p;
   @Bean
    public Person person() {
        Person p = new Person();
        p.setName("Ella");
        return p;
```

- org.scoula.main 패키지에 Main 클래스를 정의하세요.
 - ProjectConfig를 설정으로 컨텍스트 생성
 - o Person과 Parrot 빈을 추출
 - 추출된 빈의 속성 출력
 - Person's name
 - Prrot's name
 - Person's parrot

Person's parrot: null

🗹 main.Main.java

```
package org.scoula.main;
import org.scoula.beans.Parrot;
import org.scoula.beans.Person;
import org.scoula.config.ProjectConfig;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
public class Main {
    public static void main(String[] args) {
        var context = new AnnotationConfigApplicationContext(ProjectConfig.class);
        Person person = context.getBean(Person.class);
       Parrot parrot = context.getBean(Parrot.class);
        System.out.println("Person's name: " + person.getName());
       System.out.println("Parrot's name: " + parrot.getName());
        System.out.println("Person's parrot: " + person.getParrot());
       Person's name: Ella
       Parrot's name: Koko
```

- ♡ ProjectConfig를 수정하여 Person 빈의 Parrot을 설정하세요. Main을 실행하여 설정 내용을 확인하세요.
 - 메서드의 의존성 주입을 통해 Parrot을 주입하도록 설정

config.ProjectConfig.java

```
@Configuration
public class ProjectConfig {
 @Bean
  public Parrot parrot() {
    Parrot p = new Parrot();
    p.setName("Koko");
    return p;
 @Bean
  public Person person(Parrot parrot) {
    Person p = new Person();
    p.setName("Ella");
    p.setParrot(parrot);
    return p;
```

```
Person's name: Ella
Parrot's name: Koko
Person's parrot: Parrot : Koko
```

- ☑ org.scoula.beans 패키지에 Parrot2 클래스를 정의하세요.
 - 자동 등록되도록 지정
 - o name을 "Koko"로 지정

beans.Parrot2.java

```
package org.scoula.beans;
import org.springframework.stereotype.Component;
@Component
public class Parrot2 {
    private String name = "Koko";
    public String getName() {
        return name;
    public void setName(String name) {
        this.name = name;
    @Override
    public String toString() {
        return "Parrot : " + name;
```

- 🗸 org.scoula.beans 패키지에 Person2 클래스를 정의하세요.
 - 빈으로 자동 등록되도록 지정
 - o name을 "Ella"로 지정
 - o Parrot2를 오토와이어를 이용해 의존성 주입 지정

beans.Person2.java

```
@Component
public class Person2 {
 private String name = "Ella";
 @Autowired
 private Parrot2 parrot;
 public String getName() {
    return name;
 public void setName(String name) {
    this.name = name;
  public Parrot2 getParrot() {
    return parrot;
  public void setParrot(Parrot2 parrot) {
    this.parrot = parrot;
```

- 💿 org.scoula.config 패키지에 ProjectConfig2 클래스를 정의하세요.
 - 컴포넌트 스캔을 통해 자동으로 빈을 등록

config.ProjectConfig2.java

```
package org.scoula.config;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
@Configuration
@ComponentScan(basePackages = "org.scoula.beans")
public class ProjectConfig2 {
}
```

- 🗸 org.scoula.main 패키지에 Main2 클래스를 정의하세요.
 - ProjectConfig2를 설정으로 컨텍스트 생성
 - o Person2빈을 추출
 - 추출된 빈의 속성 출력
 - Person's name
 - Person's parrot

main.Main2.java

```
package org.scoula.main;
import org.scoula.beans.Person2;
import org.scoula.config.ProjectConfig2;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
public class Main2 {
    public static void main(String[] args) {
        var context = new AnnotationConfigApplicationContext(ProjectConfig2.class);
        Person2 p = context.getBean(Person2.class);
       System.out.println("Person's name: " + p.getName());
        System.out.println("Person's parrot: " + p.getParrot());
```

Person's name: Ella

Person's parrot: Parrot: Koko

- 🗸 org.scoula.beans 패키지에 Person3 클래스를 정의하세요.
 - 빈으로 자동 등록되도록 지정
 - o name을 "Ella"로 지정
 - o Parrot2를 생성자 주입을 통해 자동 설정

beans.Person3.java

```
@Component
public class Person3 {
 private String name = "Ella";
 private final Parrot2 parrot;
 @Autowired
  public Person3(Parrot2 parrot) {
    this.parrot = parrot;
 public String getName() {
    return name;
  public void setName(String name) {
    this.name = name;
  public Parrot2 getParrot() {
    return parrot;
```