

## **@Component**

- **@Component** annotation tells that an annotated class is a Spring bean or a spring component
- In this case, the spring container automatically create a spring bean (i.e. spring container creates an object to the class and manages it)

**//create a class**

**@Component**

```
public class DeptController{  
    public String getDept(){  
        return "its the Department controller";  
    }  
}
```

**main()**

```
{  
    ConfigurableApplicationContext context=SpringApplication.run(AnnotationsApplication.class,args);
```

```
    BookController ob=context.getBean(DeptController.class);  
    System.out.println(ob.getDept());  
}
```

## @Autowired

The @Autowired annotation is used to inject the bean automatically. It is used in construction injection, setter injection and field injection.

```
package net.javaguides.springannotations.service;

import org.springframework.stereotype.Component;

@Component
public class VegPizza {

    public String getPizza(){
        return "Veg Pizza!";
    }
}

import net.javaguides.springannotations.service.VegPizza;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class PizzaController {

    private VegPizza vegPizza;

    @Autowired
    public PizzaController(VegPizza vegPizza){
        this.vegPizza = vegPizza;
    }

    public String getPizza(){
        return vegPizza.getPizza();
    }
}

import ...

@SpringBootApplication
public class SpringAnnotationsApplication {

    public static void main(String[] args) {
        var context : ConfigurableApplicationContext = SpringApplication.run(SpringAnnotationsApplication.class, args);
        PizzaController pizzaController = (PizzaController) context.getBean(name: "pizzaController");
        System.out.println(pizzaController.getPizza());
    }
}
```

```

import net.javaguides.springannotations.service.VegPizza;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

@Component
public class PizzaController {

    private VegPizza vegPizza;

    // @Autowired
    // public PizzaController(VegPizza vegPizza){
    //     this.vegPizza = vegPizza;
    // }

    @Autowired
    public void setVegPizza(VegPizza vegPizza) {
        this.vegPizza = vegPizza;
    }

    public String getPizza(){
        return vegPizza.getPizza();
    }
}

```

```

@Component
public class PizzaController {

    @Autowired
    private VegPizza vegPizza;

    // @Autowired
    // public PizzaController(VegPizza vegPizza){
    //     this.vegPizza = vegPizza;
    // }

    // @Autowired
    // public void setVegPizza(VegPizza vegPizza) {
    //     this.vegPizza = vegPizza;
    // }

    public String getPizza(){
        return vegPizza.getPizza();
    }
}

```

## @Qualifier

Used in combination with @Autowired in order to avoid confusion when two or more beans configured for same type

Pizza interface

```
public interface Pizza {  
    String getPizza();  
}
```

VegPizza implementing Pizza Interface

```
import org.springframework.stereotype.Component;  
  
@Component  
public class VegPizza implements Pizza{  
    @Override  
    public String getPizza(){  
        return "Veg Pizza!";  
    }  
}
```

NonVegPizza implementing Pizza Interface

```
import org.springframework.stereotype.Component;  
  
@Component  
public class NonVegPizza implements Pizza{  
    @Override  
    public String getPizza() {  
        return "Non-veg Pizza";  
    }  
}
```

## Class which implements the qualifier

```
@Component
public class PizzaController {

    private Pizza pizza;

    @Autowired
    public PizzaController(@Qualifier("vegPizza") Pizza pizza){
        this.pizza = pizza;
    }

    // @Autowired
    // public void setVegPizza(VegPizza vegPizza) {
    //     this.vegPizza = vegPizza;
    // }

    public String getPizza(){
        return pizza.getPizza();
    }
}

import ...

@SpringBootApplication
public class SpringAnnotationsApplication {

    public static void main(String[] args) {
        var context : ConfigurableApplicationContext = SpringApplication.run(SpringAnnotationsApplication.class, args);
        PizzaController pizzaController = (PizzaController) context.getBean(name: "pizzaController");
        System.out.println(pizzaController.getPizza());
    }
}
```