

DIFFERENCE BETWEEN SETTER AND CONSTRUCTOR INJECTION , ALONG WITH QUALIFIER ANNOTATION USE.

- Create one interface :

```
➤ package com.app.service;  
  
public interface Animal {  
    String voice();  
}
```

- create one class cat which will implement interface:

```
package com.app.service.impl;  
  
import com.app.service.Animal;  
import org.springframework.stereotype.Service;  
  
@Service  
public class Cat implements Animal {  
    @Override  
    public String voice() {  
        return "meow";  
    }  
}
```

- create one more class dog which will also implement interface:

```
➤ package com.app.service.impl;  
  
import com.app.service.Animal;  
import org.springframework.stereotype.Service;  
  
@Service  
public class Dog implements Animal {  
    @Override  
    public String voice() {  
        return "bark";  
    }  
}
```

- create controller to test it:

```

package com.app.controller;

import com.app.service.Animal;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/autowire")
public class AnimalController {

    // 1: according to 1st written code it will print cat class's string, using
    // setter

    //     private Animal animal;

    //     @Autowired
    //     public void setAnimal(@Qualifier("cat") Animal animal) {
    //         this.animal = animal;
    //     }

    // 2: according to 2nd written code it will print cat class's string, using
    // constructor
    //     private Animal animal;
    //     @Autowired
    //     public AnimalController(@Qualifier("cat") Animal animal) {
    //         this.animal = animal;
    //     }

    // 3: according to 3rd written code it will not print cat class string because
    // we have removed service annotation ,
    // spring boot will not treat as bean of service class it will throw
    // UnsatisfiedDependencyException

    //     private Animal animal;
    //     @Autowired
    //     public AnimalController(@Qualifier("cat") Animal animal) {
    //         this.animal = animal;
    //     }

```

```

// 4th: according to 4th written code it will not print cat class string because
we have removed service annotation ,
// however we make required false still spring boot will not treat as bean
of service class it will throw UnsatisfiedDependencyException
//     private Animal animal;
//     @Autowired(required = false)
//     public AnimalController(@Qualifier("cat") Animal animal) {
//         this.animal = animal;
//     }

// 5th: according to 5th written code it will not print cat class string because
we have removed service annotation ,
// however we make required false still spring boot will not treat as bean
of service class but it will not throw exception at runtime , but it will not give
data
    private Animal animal;

    @Autowired(required = false)
    public void setAnimal(@Qualifier("cat") Animal animal) {
        this.animal = animal;
    }

    @GetMapping
    public String getCharacters() {
        return animal.voice();
    }
}

```

➤ now hit the url from browser: <http://localhost:8080/autowire>