9. Spring boot: Dynamically Initialized Beans | Value Annotation

- We have discuss about Unsatisfied Dependency problem.
- And Resolve it Using @Qualifier annotation.

As per discussion In Video Lecture.

Many Engineers asked the doubt that:

It will break dependency inversion

```
package com.springProject.SpringCourseProject.component;

public interface Order {
    String createOrder();
}
```

```
package com.springProject.SpringCourseProject.imple;

import com.springProject.SpringCourseProject.component.Order;
import org.springframework.stereotype.Component;

@Component
public class OfflineOrder implements Order {

    public OfflineOrder(){
        System.out.println("Offline Order Initialized.");
    }

    @Override
    public String createOrder() {
        return "Offline Order Created Successful!";
    }
}
```

```
package com.springProject.SpringCourseProject.imple;

import com.springProject.SpringCourseProject.component.Order;
import org.springframework.stereotype.Component;
```

```
@Component
public class OnlineOrder implements Order {

    public OnlineOrder()
    {
        System.out.println("OnlineOrder Order Initialized.");
    }
    @Override
    public String createOrder() {
        return "Online Order Created Successful!";
    }
}
```

```
package com.springProject.SpringCourseProject.controller;


import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api")
public class OrderController {

    @Autowired
    Order order;

    public OrderController(){
        System.out.println("OrderController Initialized.");
    }

    @PostMapping("/createOrder")
    public ResponseEntity<String> createOrder(){
        return ResponseEntity.ok(order.createOrder());
    }
}
```

****

APPLICATION FAILED TO START

****

Description:

Field order in com.springProject.SpringCourseProject.controller.OrderController required a single bean, but 2 were found:

- offlineOrder: defined in file [C:\volume e\java_projects\SpringCourseProject\target\classes\com\springProject\SpringCourseProject\imple\OfflineOrder.class]

- onlineOrder: defined in file [C:\volume e\java_projects\SpringCourseProject\target\classes\com\springProject\SpringCourseProject\imple\OnlineOrder.class]

This may be due to missing parameter name information

Action:

Consider marking one of the beans as @Primary, updating the consumer to accept multiple beans, or using @Qualifier to identify the bean that should be consumed

Ensure that your compiler is configured to use the '-parameters' flag.

You may need to update both your build tool settings as well as your IDE.

(See https://github.com/spring-projects/spring-framework/wiki/Upgrading-to-Spring-Framework-6.x#parameter-name-retention)

Process finished with exit code 1

To Solve this problem we are use @Qualifier annotations

```
package com.springProject.SpringCourseProject.imple;

import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;

@Qualifier("OfflineOrder")
@Component
public class OfflineOrder implements Order {

    public OfflineOrder(){
        System.out.println("Offline Order Initialized.");
    }

    @Override
    public String createOrder() {
        return "Offline Order Created Successful!";
    }
}
```

```
package com.springProject.SpringCourseProject.imple;

import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;

@Qualifier("OnlineOrder")
@Component
public class OnlineOrder implements Order {

    public OnlineOrder()
    {
        System.out.println("OnlineOrder Order Initialized.");
    }
    @Override
    public String createOrder() {
        return "Online Order Created Successful!";
    }
}
```

```
package com.springProject.SpringCourseProject.controller;


import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api")
public class OrderController {

    @Qualifier("OnlineOrder")
    @Autowired
    Order order;

    public OrderController(){
        System.out.println("OrderController Initialized.");
    }

    @PostMapping("/createOrder")
    public ResponseEntity<String> createOrder(){
        return ResponseEntity.ok(order.createOrder());
```

```
    }
}
```

OrderController Initialized.

OnlineOrder Order Initialized.

Offline Order Initialized.

2024-08-21T08:11:07.951+05:30 INFO 2960 --- [SpringCourseProject] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'

2024-08-21T08:11:07.967+05:30 INFO 2960 --- [SpringCourseProject] [main] c.s.S.SpringCourseProjectApplication   : Started SpringCourseProjectApplication in 4.046 seconds (process running for 4.862)

1. **Hitting API** http://localhost:8080/api/createOrder

Online Order Created Successful!

PROBLEM

- When we are using @Qualifier we hardcoded the value which static and it's break the law of dependency Inversion Control.
- **Dependency Inversion Says We can always provide dynamic value.**
- If User Wants OfflineOrder they cannot be able to use it.

1. **Solution Using @Qualifier**

Industry Using It NOW DAYS

```
package com.springProject.SpringCourseProject.controller;


import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api")
public class OrderController {
```

```
    @Qualifier("OnlineOrder")
    @Autowired
    Order onlineOrder;

    @Qualifier("OfflineOrder")
    @Autowired
    Order offlineOrder;

    public OrderController(){
        System.out.println("OrderController Initialized.");
    }

    @PostMapping("/createOrder")
    public ResponseEntity<String> createOrder(@RequestParam boolean isOnlineOrder){

        String msg = isOnlineOrder ? onlineOrder.createOrder() :
offlineOrder.createOrder();
        return ResponseEntity.ok(msg);
    }
}
```

Hitting API http://localhost:8080/api/createOrder?isOnlineOrder=false

Offline Order Created Successful!

2. **Solution Using @Bean and @Value:**

@Value:

It is used to inject values from various sources like property file, environment variables or inline literals.

```
package com.springProject.SpringCourseProject.imple;

import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;


public class OfflineOrder implements Order {

    public OfflineOrder(){

        System.out.println("Offline Order Initialized.");
    }
```

```
    @Override
    public String createOrder() {
        System.out.println( "Offline Order Created Successful!");
        return "Offline Order Created Successful!";
    }
}
```

```
package com.springProject.SpringCourseProject.imple;

import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;


public class OnlineOrder implements Order {

    public OnlineOrder()
    {
        System.out.println("OnlineOrder Order Initialized.");
    }
    @Override
    public String createOrder() {
        System.out.println( "Online Order Created Successful!");
        return "Online Order Created Successful!";
    }
}
```

```
//Application.properties
spring.application.name=SpringCourseProject
isOnlineOrder=false
```

```
package com.springProject.SpringCourseProject.config;

import com.springProject.SpringCourseProject.component.Order;
import com.springProject.SpringCourseProject.imple.OfflineOrder;
import com.springProject.SpringCourseProject.imple.OnlineOrder;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class AppConfig {
    @Bean
```

```
    public Order createOrderBean(@Value("${isOnlineOrder}") boolean isOnlineOrder){
        return isOnlineOrder ? new OnlineOrder():new OfflineOrder();
    }
}
```

```
package com.springProject.SpringCourseProject.controller;


import com.springProject.SpringCourseProject.component.Order;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api")
public class OrderController {
    @Autowired
    Order order;

    public OrderController(){
        System.out.println("OrderController Initialized.");
    }

    @PostMapping("/createOrder")
    public ResponseEntity<String> createOrder(){
        String msg = order.createOrder();
        return ResponseEntity.ok(msg);
    }
}
```

Output:

OrderController Initialized.

Offline Order Initialized.

Hitting API http://localhost:8080/api/createOrder

Offline Order Created Successful!