

الاسم: حسام عابد الصاعدي

في هذه المثل الاول:

تم استخدام تعليمة @Bean حيث تستخدم في هذ المثال لتعريف دالة "getMessage1" في حاوية spring

كما تم بناء داله اسمها "getMessage1" من نوع نصي تقوم بعمليتين

- العملية الأولى يتم طباعة جملة نصية "Hey from Message1"
- العملية الثانية يتم ارجاع قيمة وهي قيمة نصية "1"

لذاك عن تنفيذ الكود سوف يتم طباعة الجملة النصية التالية: Hey from Message1 وارجاع قيمة "1" في الحاوية

```
1 usage
@SpringBootApplication
public class SpringPollApplication {

    public static void main(String[] args) { SpringApplication.run(SpringPollApplication.class, args); }

    @Bean
    public String getMessage1(){
        System.out.println("hey from message1");
        return "1";
    }
}
```

في هذه المثال الثاني:

تم استخدام تعليمة @Bean حيث تستخدم في هذ المثال لتعريف دالة "getMessage1" ودالة "getMessage2" في حاوية spring

وتم استخدام تعليمة Qualifier("1") لتمييز الدالة واعطائها الأولوية في التنفيذ من نفس النوع

كما تم بناء الدالتين "getMessage1" و "getMessage2"

الدالة: "getMessage1" تقوم بعمليتين

- العملية الأولى طباعة جملة نصية "hey from message1"
- العملية الثانية ارجاع القيمة نصية "1"

الدالة: "getMessage2" تقوم بثلاث عمليات

- العملية الأولى: استقبال القيمة من الحاوية وهي "1" المرجعه من الدالة "getMessage1"
- العملية الثانية: طباعة جملة نصية "hey from message2"
- العملية الثالثة: تقوم بارجاع قيمة المتغير data وهيا "1"

لذاك عن تنفيذ الكود سوف يتم طباعة الجملة النصية التالية:

Hey from Message1 وارجاع قيمة "1" في الحاوية

Hey from Message2 وارجاع قيمة data في الحاوية

```
1 usage
2 @SpringBootApplication
3 public class SpringPollApplication {
4
5     public static void main(String[] args) { SpringApplication.run(SpringPollApplication.class, args); }
6
7     @Bean
8     @Qualifier("1")
9     public String getMessage1(){
10         System.out.println("hey from message1");
11         return "1";
12     }
13
14     @Bean
15     public String getMessage2(@Qualifier("1") String data ){
16         System.out.println("hey from message2");
17         return data ;
18     }
19 }
```

في هذه المثال الثالث:

تم استخدام تعليمة @Bean حيث تستخدم في هذ المثال لتعريف دالة "getMessage1" ودالة "getMessage2" و  
spring في حاوية getMessage2

وتم استخدام تعليمة Qualifier() ل تمييز وتحديد دالة في التنفيذ من نفس النوع

كما تم بناء ثلاثة دوال getMessage1 و getMessage2 و getMessage3

الدالة: getMessage1 تقوم بعمليتين

- العملية الأولى يتم طباعة جملة نصية "hey from message1"
- العملية الثانية ارجاع القيمة نصية "1"

الدالة: getMessage2 تقوم بثلاثة عمليات

- العملية الأولى: استقبال القيمة من الحاوية وهي "3" المرجعة من الدالة getMessage3 بناء على الأفضلية من Qualifier("3")
- العملية الثانية: يتم طباعة جملة نصية "hey from message2"
- العملية الثالثة: ارجاع القيمة النصية data

الدالة: "getMessage3" تقوم بعمليتين

- العملية الأولى: يتم طباعة جملة نصية "hey from message3"
- العملية الثانية: ارجاع القيمة النصية "3"

لذاك عن تنفيذ الكود سوف يتم طباعة الجملة النصية التالية:

Hey from Message1

Hey from Message3

Hey from Message2

```
@Bean
@Qualifier("1")
public String getMessage1(){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}
```

في هذه المثال الرابع:

تم استخدام تعليمة @Bean حيث تستخدم في هذا المثال لتعريف دالة "getMessage1" ودالة "getMessage2" و  
spring في حاوية getMessage2

وتم استخدام تعليمة Qualifier() لتمييز وتحديد دالة في التنفيذ من نفس النوع

كما تم بناء ثلاثة دوال getMessage1 و getMessage2 و getMessage3

الدالة: getMessage1 تقوم بعمليتين

- العملية الأولى يتم طباعة جملة نصية "hey from message1"
- العملية الثانية ارجاع القيمة نصية "1"

الدالة: getMessage2 تقوم بثلاثة عمليات

- العملية الأولى: استقبال القيمة من الحاوية وهي "3" المرجعة من الدالة getMessage3 بناء على الأفضلية من Qualifier ("3")
- العملية الثانية: يتم طباعة جملة نصية "hey from message2"
- العملية الثالثة: ارجاع القيمة النصية data

الدالة: "getMessage3" تقوم بعمليتين

- العملية الأولى: يتم طباعة جملة نصية "hey from message3"
- العملية الثانية: ارجاع القيمة النصية "3"

كما تم انشاء كلاس باسم MainController و Constructor يحتوي على Qualifier ("2") ومتغير من نوع نصي يتقبل القيمة  
من الدالة getMessage2

لذاك عن تنفيذ الكود سوف يتم طباعة الجملة النصية التالية:

Hey from Message3 يتم طباعة أولاً لأن (getMessage3) Bean "3" يتم إنشاؤه قبل Bean الأخرى

Hey from Message2 بعد ذلك، يتم استدعاء getMessage2، والتي تعتمد على Bean "3"

hey from Main Controller يتم ينشئ Spring كائنًا من MainController، ويستدعي المنشئ الخاص به.

Hey from Message1 أخيرًا، يتم إنشاء (getMessage1) Bean "1".

في هذه المثال الخامسة:

تم استخدام تعليمة @Bean حيث تستخدم في هذا المثال لتعريف دالة "getMessage1" ودالة "getMessage2" و  
getMessage2 في حاوية spring

وتم استخدام تعليمة Qualifier() لتمييز وتحديد الدالة في التنفيذ من نفس النوع

كما تم بناء ثلاثة دوال getMessage1 و getMessage2 و getMessage3

الدالة: getMessage1 تقوم بثلاثة عمليات

- العملية الأولى يتم تعرفي كان من نوع MainController
- العملية الثانية يتم طباعة جملة نصية "hey from message1"
- العملية الثانية ارجاع القيمة نصية "1"

الدالة: getMessage2 تقوم بثلاثة عمليات

- العملية الأولى: استقبال القيمة من الحاوية وهي "3" المرجعة من الدالة getMessage3 بناء على الأفضلية من Qualifier ("3")
- العملية الثانية: يتم طباعة جملة نصية "hey from message2"
- العملية الثالثة: ارجاع القيمة النصية data

الدالة: "getMessage3" تقوم بعمليتين

- العملية الأولى: يتم طباعة جملة نصية "hey from message3"
- العملية الثانية: ارجاع القيمة النصية "3"

كما تم انشاء كلاس باسم MainController و Constructor يحتوي على Qualifier ("2") ومتغير من نوع نصي يتقبل القيمة  
من الدالة getMessage2

لذاك عن تنفيذ الكود سوف يتم طباعة الجملة النصية التالية:

Hey from Message3 يتم طباعة أولاً لأن (getMessage3) Bean "3" يتم إنشاؤه قبل Bean الأخرى

Hey from Message2 بعد ذلك، يتم استدعاء getMessage2، والتي تعتمد على Bean "3"

hey from Main Controller يتم ينشئ Spring كائنًا من MainController، ويستدعي المنشئ الخاص به.

Hey from Message1 أخيرًا، يتم إنشاء (getMessage1) Bean "1" ،.

```

@Bean
@Qualifier("1")
public String getMessage1(){
    System.out.println("hey from message1");
    return "1";
}

@Bean
@Qualifier("2")
public String getMessage2(@Qualifier("3") String data ){
    System.out.println("hey from message2");
    return data;
}

@Bean
@Qualifier("3")
public String getMessage3(){
    System.out.println("hey from message3");
    return "3" ;
}

```

```

@Component
public class MainController {

    1 usage
    String data;

    public MainController(@Qualifier("1") String data){
        this.data=data;
        System.out.println("hey from Main controller");
    }
}

```

في هذه المثال الخامس:

تم استخدام تعليمة @Bean حيث تستخدم في هذا المثال لتعريف دالة "getMessage1" ودالة "getMessage2" و  
spring في حاوية getMessage2

وتم استخدام تعليمة Qualifier("3") لتمييز وتحديد الدالة في التنفيذ من نفس النوع  
كما تم بناء ثلاثة دوال getMessage1 و getMessage2 و getMessage3

الدالة: getMessage1 تقوم بعمليتين

- العملية الأولى يتم طباعة جملة نصية "hey from message1"
- العملية الثانية ارجاع القيمة نصية "1"

الدالة: getMessage2 تقوم بثلاثة عمليات

- العملية الأولى: استقبال القيمة من الحاوية وهي "3" المرجعة من الدالة getMessage3 بناء على الأفضلية من Qualifier ("3")
- العملية الثانية: يتم طباعة جملة نصية "hey from message2"
- العملية الثالثة: ارجاع القيمة النصية data

الدالة: "getMessage3" تقوم بعمليتين

- العملية الأولى: يتم طباعة جملة نصية "hey from message3"
- العملية الثانية: ارجاع القيمة النصية "3"

كما تم انشاء كلاس باسم MainController و Constructor يحتوي على Qualifier ("2") ومتغير من نوع نصي يتقبل  
القيمة من الدالة getMessage2

لذاك عن تنفيذ الكود سوف يتم طباعة الجملة النصية التالية:

Hey from Message3 يتم طباعة أولاً لأن (getMessage3) "3" Bean يتم إنشاؤه قبل Bean الأخرى

Hey from Message2 بعد ذلك، يتم استدعاء getMessage2، والتي تعتمد على "3" Bean

hey from Main Controller يتم ينشئ Spring كائنًا من MainController، ويستدعي المنشئ الخاص به.

Hey from Message1 أخيرًا، يتم إنشاء (getMessage1) "1" Bean ،.



```

15
16 @Bean
17 @Qualifier("1")
18 public String getMessage1(MainController mainController){
19     System.out.println("hey from message1");
20     return "1";
21 }
22
23 @Bean
24 @Qualifier("2")
25 public String getMessage2(@Qualifier("3") String data ){
26     System.out.println("hey from message2");
27     return data;
28 }
29
30 @Bean
31 @Qualifier("3")
32 public String getMessage3(){
33     System.out.println("hey from message3");
34     return "3" ;
35 }

```

```

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Component;

1 usage
@Component
public class MainController {

    1 usage
    String data;

    public MainController(@Qualifier("2") String data){
        this.data=data;
        System.out.println("hey from Main controller");
    }
}

```