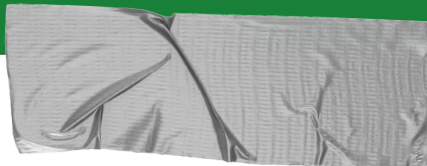

Spring Certified #2



A question lead guide to prepare Spring certification



Configuration

What are some valid methods for adding a Bean definition to the IoC Container in Spring? (select 2)

- Using the **@Bean** annotation on a method in a configuration class.
- Using XML-based configuration
- Using the **@ComponentScan** annotation
- Using the new operator

answer

1. Using the **@Bean** annotation on a method in a configuration class: This method returns an object that should be added as a bean to the application context.
2. Using XML-based configuration: The **<bean>** element can be used to define a bean in an XML file that is loaded by the application context.

Incorrect answers:

1. Using the **@ComponentScan** annotation: While this annotation does scan for classes with **@Component** annotations and adds them to the container, it doesn't provide a way to add a bean definition explicitly.
2. Using the **new** operator: Instantiating objects using **new** doesn't add them to the application context and they won't be available for dependency injection.



Data Management

What comparison operator can be used at the end of a JPA Repository finder method to generate a valid query? (select 2)

- **LessThan**
- **Is**
- **BiggerThan**
- **DifferentOf**

answer

Two correct answers:

1. **LessThan**: This operator is used to find entities where the specified attribute is less than the given value.
2. **Is**: This operator is used to find entities where the specified attribute is equal to the given value.



Spring MVC

Which of the following handler method signatures are valid in Spring MVC? (select 2)

- `@RequestMapping(value="/orders")`
`public String getOrderWeight(@RequestParam String orderId)`
- `@PutMapping(value="/orders/{orderId}")`
`public Order createOrder(@PathVariable("orderId") int orderId
, @RequestBody Order order)`
- `@PostMapping("/orders")`
`public ResponseEntity createOrder(@RequestParam Order
order)`
- `@GetMapping("/orders/{id}")`
`public User getOrder(@RequestBody id, Model m)`

answer

```
@RequestMapping(value="/orders")
```

```
public String getOrderWeight(@RequestParam String orderId)
```

This handler method specifies a request mapping for "/orders" and has a valid parameter of **@RequestParam**.

```
@PutMapping(value="/orders/{orderId}")
```

```
public Order createOrder(@PathVariable("orderId") int orderId, @RequestBody Order order)
```

This handler method specifies a request mapping for "orders/{orderId}" and has valid parameters of **@PathVariable** and **@RequestBody**.

```
@PostMapping("/order")
```

```
public ResponseEntity createOrder(@RequestParam Order order)
```

This handler method has an incorrect parameter of **@RequestParam** with a non-primitive type.

```
@GetMapping("/orders/{orderId}")
```

```
public User getOrder(@RequestBody orderId, Model m)
```

This handler method has an incorrect parameter of **@RequestBody**, which is not compliant with the URL parameter.

—



<https://bit.ly/2v7222>