Spring Certified #8

A question lead guide to prepare Spring certification

Security

To enable annotation-based security, we need to add the _____ annotation on any @Configuration class.

- ➔ **@EnableMethodSecurity**
- ➔ **@SpringSecurity**
- ➔ **@SecuredApplication**
- ➔ **@EnableApplicationSecurity**

@EnableMethodSecurity

To enable annotation-based security, we need to add the `@EnableMethodSecurity` annotation on any `@Configuration` class. This is how our configuration class will look like:

```
package com.javadevjournal.config;
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity ;
/**
 * EnableMethodSecurity to allow method level Spring security annotation for our application
 */
@EnableMethodSecurity ( securedEnabled = true,  jsr250Enabled = true,     prePostEnabled = true)
public class MethodSecurityConfig {
    //default configuration class
}
```

Let's look at a few important parameters of the `@EnableMethodSecurity` annotation

- **`securedEnabled`** – Determine if the `@Security` annotation should be enabled.
- **`jsr250Enabled`** – Allow us to use JSR250-based annotation (e.g. `@RoleAllowed`).
- **`prePostEnabled`** – Enable Spring's pre/post annotations.

https://www.javadevjournal.com/spring-security/spring-method-security/

https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/config/annotation/method/configuration/EnableMethodSecurity.html

Data Management

In Spring Data, ______transactions are bound to a single database connection

- ➔ **local**
- ➔ **global**
- ➔ **internal**
- ➔ **external**

local

In Spring Data**, local transactions are bound to a single database connection**, whereas global transactions can span multiple database connections or even multiple transactional resources (such as JMS or JCA resources).

Local transactions are managed using the `@Transactional` annotation or the `TransactionTemplate` class, and are suitable for operations that involve only one database connection. Global transactions, on the other hand, are managed by a transaction manager that coordinates transactions across multiple resources, such as JTA (Java Transaction API) or Atomikos.

In summary, local transactions are simpler and more lightweight than global transactions, but cannot be used for distributed transactions that involve multiple resources. Global transactions provide more robustness and flexibility but require more complex configurations and are generally slower.

Local transactions are resource-specific, such as a transaction associated with a JDBC connection. Local transactions may be easier to use, but have significant disadvantages: they cannot work across multiple transactional resources.

https://docs.spring.io/spring-framework/reference/data-access/transaction/motivation.html

Spring MVC

Which method parameter in the given code is a likely candidate for the @RequestParam annotation in Spring?

```
@RequestMapping("orders/{orderId}/dispatch-center/{centerId}")
public String someHandlerMethod(String orderId, String centerId,
String contractorId) {
    // functional code
}
```

- ➔ **orderId**
- ➔ **centerId**
- ➔ **contractorId**

contractorId

@interface **RequestParam**

Annotation which indicates that a method parameter should be bound to a web request parameter.

https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/RequestParam.html

In the question, only `contractorId` is not present in the URL path ("`orders/{orderId}/dispatch-center/{centerId}`"). So, it could be mapped to a RequestParam.

https://bit.ly/2v7222

https://spring-book.mystrikingly.com