Spring Professional Exam Tutorial v5.0

Question 12

Question 12 - In which security annotation are you allowed to use SpEL?

Spring Security supports SpEL expressions in following annotations:

- @PreAuthorize

```
@PreAuthorize("hasRole('ROLE_EMPLOYEES_CREATE') || 'TEST'.equals(#employee.getFirstName())")
```

- @PostAuthorize

```
@PostAuthorize("hasRole('ROLE_CUSTOMERS_QA') && returnObject.firstName.equals('TEST')")
```

- @PreFilter

```
@PreFilter("hasRole('ROLE_CUSTOMERS_QA') && filterObject.firstName.equals('TEST')")
```

- @PostFilter

```
@PostFilter("hasRole('ROLE_CUSTOMERS_QA') && filterObject.firstName.equals('TEST')")
```

Question 12 - In which security annotation are you allowed to use SpEL?

Main difference between `@PreAuthorize / @PostAuthorize` and `@PreFilter / @PostFilter` annotations is that `@PreAuthorize / @PostAuthorize` are used to create expression that will check if method can be executed, `@PreFilter / @PostFilter` on the other hand are used to filter collections based on security rules.

Filtering Example:

```
@PostFilter("hasRole('ROLE_CUSTOMERS_QA') && filterObject.firstName.equals('TEST')")
Iterable<Customer> findAll();
```

Method execution security example:

```
@PreAuthorize("hasRole('ROLE_CUSTOMERS_READ')")
@GetMapping("/customers")
public ModelAndView index() {
    ...
}
```