Spring Professional Exam Tutorial v5.0

Question 09

Question 09 - What does @RequestMapping do?

`@RequestMapping` allows you to specify conditions that request has to match for a method to be used as request handler. `@RequestMapping` can be used at class or method level, when used at the class level, all method level mappings inherit this primary mapping, narrowing it to a specific handler method.

For example, below controllers are supposed to map `GET /say/hello requests`, even though request mapping is defined differently, all are equal.

```
@RestController
@RequestMapping("/say/hello")
public class HelloController {

    @RequestMapping(method = GET)
    public ResponseEntity<String> sayHello() {
        return ResponseEntity.ok()
                .body("Hello");
    }
}
```

```
@RestController
@RequestMapping(path = "/say", method = GET)
public class HelloController {

    @RequestMapping("/hello")
    public ResponseEntity<String> sayHello() {
        return ResponseEntity.ok()
                .body("Hello");
    }
}
```

```
@RestController
public class HelloController {

    @RequestMapping(path = "/say/hello", method = GET)
    public ResponseEntity<String> sayHello() {
        return ResponseEntity.ok()
                .body("Hello");
    }
}
```

Question 09 - What does @RequestMapping do?

`@RequestMapping` annotation allows you to specify following criteria for request:

- `path` - uri path/paths for request, for example `/api/books`
- `method` - supported HTTP method/methods: `GET, POST, HEAD, OPTIONS, PUT, PATCH, DELETE, TRACE`
- `params` - required parameters of request, for example `key1=value1, key2!=value2, key1, !key1`
- `headers` - header needs to match specified condition, for example `header1=value1, header2!=value2, header1, !header1, content-type=text/*`
- `consumes` - media types that can be consumed by request, for example `application/json`
- `produces` - media types that are produced by method handling the request, for example `application/pdf`

Question 09 - What does @RequestMapping do?

Spring also supports composed annotations for request mapping:

- `@GetMapping`
- `@PostMapping`
- `@PutMapping`
- `@PatchMapping`
- `@DeleteMapping`

Each of those annotations allows you to specify same conditions as `@RequestMapping` except for HTTP `method` field, following fields in `@*Mapping` are aliases to `@RequestMapping`: `path, params, headers, consumes, produces.`

In most of the cases it is possible to translate mappings between those annotations, one example when this is not possible is when creating `HTTP HEAD` request mapping.

```
@RestController
public class HelloController {

    @RequestMapping(path = "/say/hello", method = GET)
    public ResponseEntity<String> sayHello() {
        return ResponseEntity.ok()
                .body("Hello");
    }
}
```

```
@RestController
public class HelloController {

    @GetMapping(path = "/say/hello")
    public ResponseEntity<String> sayHello() {
        return ResponseEntity.ok()
                .body("Hello");
    }
}
```