Questions from Spring MVC lecture

1. Как анотираме контролер класовете си?

2. Как използваме @RequestMapping анотацията?

3. Има ли разлики между @GetMapping и @RequestMapping с метод Get върху метод?

4. Как може да вземем POST параметър?

5. Как предаваме обект съм Thymeleaf view?

6. Как може да вземем optional параметър?

7. Как може да зададем default стойност на параметър?

8. Каква разликата между Path variable и Query parameter?

1. Как анотираме контролер класовете си?

**@Controller Annotation**

1. Как използваме @RequestMapping анотацията?

The @RequestMapping annotation can be applied to class-level and/or method-level in a controller. The class-level annotation maps a specific request path or pattern onto a controller. You can then apply additional method-level annotations to make mappings more specific to handler methods.

Анотацията @RequestMapping може да се приложи към ниво на клас и/или ниво на метод в контролер. Анотацията на ниво клас мапва специфичен път или модел на заявка върху контролер. След това можете да приложите допълнителни анотации на ниво метод, за да направите съпоставянията по-специфични за методите на манипулатора.

Resource: https://www.geeksforgeeks.org/spring-requestmapping-annotation-with-example/

1. Има ли разлики между @GetMapping и @RequestMapping с метод Get върху метод?

Проблемът при използване на @RequestMapping е, че приема всички видове методи на заявка (get, post, put, delete, head, patch)

**@RequestMapping(value="/home", method=RequestMethod.GET)**

**public String home() {**

**return "home-view";**

**}**

И двете правят едно и също

**@GetMapping("/home")**

**public String home() {**

**return "home-view";**

**}**

1. Как може да вземем POST параметър?

С помощта на ДТО – байнинг модел също и с RequestParam annotation (*@RequestParam*) .

Simply put, we can use @RequestParam to extract query parameters, form parameters, and even files from the request.

Казано по-просто, можем да използваме @RequestParam за извличане на параметри на заявка, параметри на формуляра и дори файлове от заявката.

A Simple Mapping.

Let's say that we have an endpoint / api / foos that takes a query parameter called  *id*:

Да кажем, че имаме крайна точка / api / foos, която приема параметър на заявка, наречен id:

@GetMapping("/api/foos")

@ResponseBody

**public** String **getFoos**( @RequestParam String id) {

**return** "ID: " + id;

}

In this example , we used @RequestParam to extract the id query parameter.

A simple GET request would invoke getFoos:

В този пример използвахме @RequestParam за извличане на параметъра на заявката за идентификация.

Една проста GET заявка би извикала getFoos:

<http://localhost:8080/spring-mvc-basics/api/foos?id=abc>

----

ID: abc

In the previous example, both the variable name and the parameter name are the same.

Sometimes we want these to be different, though. Or, if we aren't using Spring Boot, we may need to do special compile-time configuration or the parameter names won't actually be in the bytecode.

Fortunately, we can configure the @RequestParam name using the name attribute:

В предишния пример и името на променливата, и името на параметъра са еднакви.

Понякога обаче искаме те да са различни. Или, ако не използваме Spring Boot, може да се наложи да направим специална конфигурация по време на компилиране или имената на параметрите всъщност няма да са в байт-кода.

За щастие можем да конфигурираме името @RequestParam с помощта на атрибута name:

@PostMapping("/api/foos")

@ResponseBody

**public** String **addFoo**( @RequestParam(name = "id") String fooId, @RequestParam String name) {

**return** "ID: " + fooId + " Name: " + name;

}

We can also do @RequestParam(value = “id”) or just  @RequestParam(“id”).

Можем също да направим @RequestParam(value = "id") или просто @RequestParam("id").

More on: <https://www.baeldung.com/spring-request-param>

1. Как предаваме обект към Thymeleaf view?

Чрез обектите които инжектираме – Model, model map, model and view, @ModelAttribute

More : <https://www.thymeleaf.org/doc/articles/springmvcaccessdata.html>

And more for Thymeleaf: <https://www.thymeleaf.org/doc/tutorials/2.1/thymeleafspring.html>

1. Как може да вземем optional параметър?

Using “required = false” in @RequestParam

**@GetMapping("/search")**

**public String search(@RequestParam(name="sort",**

**required = false) String sort) {**

**...**

**}**

1. Как може да зададем default стойност на параметър?

We can also set a default value to the @RequestParam by using the defaultValue attribute:

@GetMapping("/api/foos")

@ResponseBody

**public** String **getFoos**(@RequestParam(defaultValue = "test")

String id) {

**return** "ID: " + id;

}

1. Каква разликата между Path variable и Query parameter?

@RequestParam and @PathVariable can both be used to extract values from the request URI, but they are a bit different.

While @RequestParams extract values from the query string,  @PathVariables extract values from the URI path:

@GetMapping("/foos/{id}")

@ResponseBody

**public** String **getFooById**( @PathVariable String id) {

**return** "ID: " + id;

}

Then we can map based on the path:

http://localhost:8080/spring-mvc-basics/foos/abc

----

ID: abc

And for @RequestParam, it will be:

@GetMapping("/foos")

@ResponseBody

**public** String **getFooByIdUsingQueryParam**( @RequestParam

String id) {

**return** "ID: " + id;

}

which would give us the same response, just a different URI:

http://localhost:8080/spring-mvc-basics/foos?id=abc

----

ID: abc