Anotações no controller

Certified Developer
The Ultimate Tech Degree

DigitalHouse >
Coding School

Índice

1 Múltiplos parâmetros com o método GET

2 Múltiplos parâmetros com o método POST

1 Múltiplos parâmetros com o método GET

@GetMapping

O controller é nosso artefato central de nossos serviços REST.

Em um método dentro do controller podemos receber tantos parâmetros forem necessários;

Como por exemplo em um método **GetPerson**, se você deseja obter por parâmetro além do nome, também o sobrenome, faríamos o seguinte:

```
@RestController
public class PersonController {
   @GetMapping(path = "{name}/{lastname}")
   public String sayHello( @PathVariable String name,
                           @PathVariable String lastName) {
       return "Olá, " + name + " " + lastName;
   }
}
```

@PathVariable

É uma anotação que permite extrair informações que fazem parte da estrutura da URI, mas que não são tratados como um parâmetro.

Como o próprio nome diz, é “variável”, portanto o valor inserido na URI será aquele atribuído à variável mapeada com esta anotação.

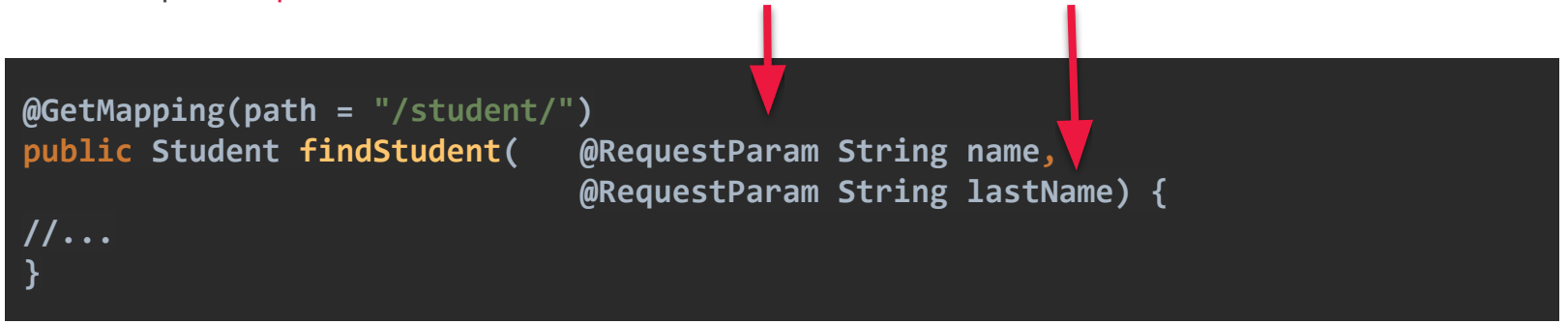
```
@GetRequest("/user/{userId}")
public User getUser(@PathVariable("userId") String userId){
//...
}
```

@RequestParam

É uma anotação que permite receber parâmetros de uma rota através do método GET, para trabalhar com eles e até poder emitir uma resposta que depende dos parâmetros obtidos.

Cada um dos parâmetros geralmente é colocado na URL após um ponto de interrogação "?" e estão delimitados por um "&".

Por exemplo: http://localhost:8080/student?name=Marcos&lastName=Silva



```
@GetMapping(path = "/student/")
public Student findStudent(    @RequestParam String name,
                               @RequestParam String lastName) {
//...
}
```

@PathVariable vs @RequestParam

@PathVariable

- Usado para recuperar valores da própria URI.
- **Formato de solicitação:**
  http://localhost:8080/employee/Marcos/Silva

```
@GetRequest("/employee/{name}/{lastname}")
public Employee getEmployee(@PathVariable("name") String name
                            @PathVariable("lastname") String
lastName) {
   return new Employee(name, lastName);
}
```

@RequestParam

- Usado para recuperar parâmetros de consulta. Você os obtém identificando após o caractere '?' na URL
- **Formato de solicitação:**
  http://localhost:8080/employee?name=Marcos&lastName=Silva

```
@GetMapping(path = "/employee/")
public Student findEmployee(@RequestParam String name,
                            @RequestParam String lastName) {

   return new Employee(name, lastName);

}
```

2 Múltiplos parâmetros com o método POST

@PostMapping

A anotação @PostMapping é usada para mapear solicitações HTTP POST para métodos do controller. É uma alternativa para:

@RequestMapping (method = RequestMethod.POST)

Os métodos com a anotação @PostMapping tratam as solicitações HTTP POST que correspondem a uma determinada URI.

Exemplo: em nossa classe Employee, criamos um método que adiciona um novo funcionário.

```
@RestController
@RequestMapping("/")
public class EmployeeRestController {
   @PostMapping(path = "/employee")
   public void addEmployee(@RequestBody Employee employee) {
       //...
   }
}
```

Payload

Como parte de uma solicitação POST ou PUT, pode-se enviar uma carga útil (mais conhecida pela palavra em inglês payload) de dados para o servidor no corpo da solicitação.

O conteúdo do corpo pode ser qualquer objeto JSON válido, por exemplo:

```
{
   "firstname": "Marcos"
   "lastname" : "Silva",
   "username" : "msilva",
   "email"    : "msilva@digitalhouse.com"
 }
```

@RequestBody

É usado para vincular uma solicitação HTTP (HTTP request) a um objeto em um parâmetro de um método de nosso controller.

Essa anotação mapeia o corpo da solicitação HTTP para um objeto de domínio, permitindo a deserialização automática do corpo da solicitação HTTP recebida para um objeto Java.

Exemplo: temos uma classe Employee e criamos um método que recebe um objeto usando a anotação @RequestBody.

```
public class Employee{
   private Long id;
   private String name;
   private String lastName;
}
```

```
@RestController
@RequestMapping("/")
public class HelloRestController {
   @PostMapping(path = "/employee")
   public void handle(@RequestBody Employee employee){
       //...
   }
}
```

@ResponseBody

Esta anotação é usada para indicar o conteúdo de uma resposta HTTP (response) em seu corpo.

Uma resposta HTTP não pode conter objetos Java, então @ResponseBody se encarrega de transformar objetos para o formato JSON ou XML.

Exemplo: o método getOrders retorna uma lista de pedidos. @ResponseBody se encarrega de transformar essa lista de objetos em JSON.

```
@GetMapping(path = "/orders/")
@ResponseBody
public List<Order> getOrders(){
   return orderService.getAllOrders();
}
```


DigitalHouse>
Coding School