

SpringMVC RequestMapping for GET parameters

How to make the RequestMapping to handle GET parameters in the url? For example i have this url

```
http://localhost:8080/userGrid?
_search=false&nd=1351972571018&rows=10&page=1&sidx=id&sord=desc
```

(from jqGrid)

how should my RequestMapping look like? I want to get the parameters using HttpRequest

Tried this:

```
@RequestMapping("/userGrid")
public @ResponseBody GridModel getUsersForGrid(HttpServletRequest request)
```

but it doesn't work.

java spring spring-mvc

asked Nov 3 '12 at 20:12



Evgeni Dimitrov

9,444 19 66 98

6 Answers

Use [@RequestParam](#) in your method arguments so Spring can bind them, also use the [@RequestMapping.params](#) array to narrow the method that will be used by spring. Sample code:

```
@RequestMapping("/userGrid",
params = {"_search", "nd", "rows", "page", "sidx", "sort"})
public @ResponseBody GridModel getUsersForGrid(
@RequestParam(value = "_search") String search,
@RequestParam(value = "nd") int nd,
@RequestParam(value = "rows") int rows,
@RequestParam(value = "page") int page,
@RequestParam(value = "sidx") int sidx,
@RequestParam(value = "sort") Sort sort) {
// Stuff here
}
```

This way Spring will only execute this method if ALL PARAMETERS are present saving you from null checking and related stuff.

edited Aug 16 '17 at 17:39

answered Nov 3 '12 at 20:28



ElderMael

5,156 4 22 49

- 5 What if I don't want all of them every time, can they be optional? Also, what if I have a lot of params? Can I just create a class with params and put the class as `@RequestParam`? Thanks! – [Djordje Ivanovic](#) May 28 '13 at 13:08
- 1 see this post stackoverflow.com/questions/4904092/... to know about optional parameters – [Thomas](#) Sep 13 '13 at 11:55
- 1 Does this work? How should the Sort object look like in the URL? I tried with the `?object[key]=value` format and it didn't work – [ViniciusPires](#) Feb 10 '15 at 13:38
- 2 To make parameters optional in newer versions of Spring (4+) use `required=false` and `Optional<T>` as parameter type e.g. `@RequestParam(value="sort",required=false) Optional<String> sort` – [csharpfolk](#) Nov 6 '16 at 18:31

You can add `@RequestMapping` like so:

```
@RequestMapping("/userGrid")
public @ResponseBody GridModel getUsersForGrid(
    @RequestParam("_search") String search,
    @RequestParam String nd,
    @RequestParam int rows,
    @RequestParam int page,
    @RequestParam String sidx,
    @RequestParam String sord) {
```

edited Nov 3 '12 at 21:07

answered Nov 3 '12 at 20:26

Reimeus

135k 10 142 207

-
- 6 Also you don't need `value = "myname"` if the method param and request param have the same name. I upvoted this answer though. – [david](#) Nov 3 '12 at 20:33
-
- You're absolutely right! See update – [Reimeus](#) Nov 3 '12 at 20:41
-
- downvoter - care to comment? – [Reimeus](#) Sep 26 '16 at 18:37
-

This will get ALL parameters from the request. For Debugging purposes only:

```
@RequestMapping (value = "/promote", method = {RequestMethod.POST, RequestMethod.GET})
public ModelAndView renderPromotePage (HttpServletRequest request) {
    Map<String, String[]> parameters = request.getParameterMap();

    for(String key : parameters.keySet()) {
        System.out.println(key);
        String[] vals = parameters.get(key);
        for(String val : vals)
            System.out.println(" -> " + val);
    }

    ModelAndView mv = new ModelAndView();
    mv.setViewName("test");
    return mv;
}
```

answered Jun 10 '13 at 0:48

Mircea Stanciu

1,676 2 21 25

-
- Does there also exist a possibility to get all mapped params + optionally ones that are not mapped yet? E.g. to detect if a client puts in parameters that aren't used. So one could log for this a warning? – [Strinder](#) Sep 1 '15 at 10:50
-
- 1 @Strinder (Late response, but here for reference): If you have the method argument `@RequestParam(defaultValue = "") MultiValueMap<String, String>`, then you will get a `Map` where each value is a `List` of `String`s for the given `key`. You could then iterate over this to see if there are any `key`s you don't recognise. `Map<String, String>` works too, if you know for certain there won't be any repeated parameters. – [user2478398](#) Jul 8 '16 at 9:25
-

If you are willing to change your uri, you could also use `PathVariable`.

```
@RequestMapping(value="/mapping/foo/{foo}/{bar}", method=RequestMethod.GET)
public String process(@PathVariable String foo,@PathVariable String bar) {
    //Perform Logic with foo and bar
}
```

NB: The first `foo` is part of the path, the second one is the `PathVariable`

answered Oct 21 '16 at 8:33

Amanuel Nega

763 11 27

You should write a kind of template into the `@RequestMapping`:

```
http://localhost:8080/userGrid?
_search=${search}&nd=${nd}&rows=${rows}&page=${page}&sid=${sid}&sord=${sord}
```

Now define your business method like following:

```
@RequestMapping("/userGrid?
_search=${search}&nd=${nd}&rows=${rows}&page=${page}&sid=${sid}&sord=${sord}")
public @ResponseBody GridModel getUsersForGrid(
    @RequestParam(value = "search") String search,
    @RequestParam(value = "nd") int nd,
    @RequestParam(value = "rows") int rows,
    @RequestParam(value = "page") int page,
    @RequestParam(value = "sid") int sid,
    @RequestParam(value = "sort") Sort sort) {
    .....
}
```

So, framework will map `${foo}` to appropriate `@RequestParam`.

Since sort may be either asc or desc I'd define it as a enum:

```
public enum Sort {  
    asc, desc  
}
```

Spring deals with enums very well.

answered Nov 3 '12 at 20:23



AlexR

92.1k 6 87 154

4 what if the parameters change in order? – [ElderMael](#) Nov 3 '12 at 20:57

Good point, @mael. I typically used Spring MVC for Restful services where the parameters are specified into the URL itself and the order is pre-defined. I learned about `params` attribute now. – [AlexR](#) Nov 4 '12 at 16:22

This works in my case:

```
@RequestMapping(value = "/savedata",  
                params = {"textArea", "localKey", "localFile"})  
@ResponseBody  
public void saveData(@RequestParam(value = "textArea") String textArea,  
                    @RequestParam(value = "localKey") String localKey,  
                    @RequestParam(value = "localFile") String localFile) {  
}
```

answered Mar 2 '17 at 9:06



Kirill Ch

571 5 18
