

Date : 23/02/2021
Spring Boot 9AM
Mr. RAGHU

WEB MVC : Request Params

*) Passing Input/Data using URL section in key=val format is called as Request Parameters.

*) By default key and values are String type.

*) URL is case-sensitive, even key names.

Format looks like:

URL ? key=val & key=val & key=val

Example Google Search:

[https://www.google.com/search](https://www.google.com/search?q=india&rlz=1C1CHBD_enIN917IN917&oq=india&aqs=chrome..69i57j46i433j0j0i131i433j69i60j69i65.2262j0j7&sourceid=chrome&ie=UTF-8)

? q=india

& rlz=1C1CHBD_enIN917IN917

& oq=india

& aqs=chrome..69i57j46i433j0j0i131i433j69i60j69i65.2262j0j7

& sourceid=chrome

& ie=UTF-8

Other Examples in our applications (find by id,name, delete by name, edit by code..)

<http://localhost:8080/emp/find?eid=98>

<http://localhost:8080/emp/delete?ename=SAM>

<http://localhost:8080/emp/edit?aid=9865>

*) Mostly this process is used with GET type request.

It supports all types of request.

----Servlets-----

```
String val = request.getParameter("key");
```

```
// Do Parse if Required.
```

<http://....../show?sid=101>

```
//Read data
```

```
String sid = request.getParameter("sid");
```

```
//parse data
```

```
int id = Integer.parseInt(sid);
```

Syntax to Read Req Param in Web MVC:

```
@RequestParam("key") DataType localVariable
```

```
--or--
```

```
@RequestParam DataType key (localVariable Name = key name)
```

*) Supports even type conversion also.

---Codes-----

1. Controller

```
package in.nareshit.raghu.controller;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
```

```
@Controller
public class StudentController {

    //.... /show?sname=ABCD

    @GetMapping("/show")
    public String readData(
        @RequestParam("sname")String name,
        Model model
    )
    {
        model.addAttribute("name", name);
        return "StdHome";
    }
}
```

```
2. UI: StdHome.jsp
<%@ page isELIgnored="false" %>
<html>
<body>
Data is : ${name}
</body>
</html>
```

```
3. application.properties
server.port=9898
spring.mvc.view.prefix=/WEB-INF/pages/
spring.mvc.view.suffix=.jsp
```

```
Req URL#1:
http://localhost:9898/show?sname=SAM
Output: Data is : SAM
```

```
Req URL#2
http://localhost:9898/show
Output: BAD REQUEST - 400 Required String parameter 'sname' is not
present.
```

```
*) Note:
-> Every RequestParam by default required. They are not optional to
ignore.
-> We can make parameter optional by adding attribute
    required = false [default is true]

-> If we make parameter optional, then default value is null.
```

```
ex: @RequestParam(value = "sname", required = false)String name
```

```
--Core Java---
```

```
int id=-1; //valid
int id=null; //invalid (null can not be assigned to primitive type)
```

```
Integer id=null; //valid
-----
```

-> We can even modify default value from null to any other
(for optional parameters only)

```
@RequestParam(
    value = "sname",
    required = false,
    defaultValue = "NONE") String name
```

=> We can send even multiple parameters, no need of following order.
[Data binding is applied based on key name]

```
http://localhost:9898/show?sid=1234&sname=ABCD
http://localhost:9898/show?sname=ABCD&sid=1234
Output is same for above two Req URLs.
```

=> If same key found in URL multiple times with different value
then first matching combination is taken, ie key searching order.
Other values(pairs) are ignored.

```
http://localhost:9898/show?sid=10&sid=20&sid=30&sname=ABCD
Output: SID-10, SNAME-ABCD
```

=> If we send additional params which are not present in code they
are ignored, no error/no exception.

```
*) consider code has sid,sname params
Req URL:
http://localhost:9898/show?sid=10&sname=ABCD&sfee=35.0
Output: sid=10, sname=ABCD.
```

=> Some times we need to send same key with multiple values, to catch
all values use List collection variable.

```
*) Controller:
package in.nareshit.raghu.controller;
```

```
import java.util.List;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
```

```
@Controller
public class StudentController {
```

```

    @GetMapping("/mydata")
    public String getData(
        @RequestParam("sub") List<String> sub,
        Model model
    )
    {
        System.out.println(sub);
        model.addAttribute("sub", sub);
        return "StdNew";
    }
}

```

*) UI : StdNew.jsp
<html><body> Data is : \${sub} </body></html>

Ex:

http://localhost:9898/mydata?sub=ENG&sub=MAT&sub=SCI

Output: Data is : [ENG, MAT, SCI]

=> if invalid data is passed for param data type then
Bad Request - 400 Status is given.

*) MethodArgumentTypeMismatchException:
Failed to convert value of type 'java.lang.String' to required type
'int';
NumberFormatException: For input string: "ABC"

http://localhost:9898/show?sid=ABCD (wrong data)

Q) What is difference between
i) @RequestParam and @ModelAttribute?
A)

@RequestParam : To send one value / read as one value (UI->Controller)

@ModelAttribute: To send bulk(FORM DATA) and read as one object
(UI--->Controller)

ii) Model and ModelAttribute?

b) Model : To send data from Controller ---> UI

ModelAttribute: To send data from UI ---> Controller

*) If any primitive type is declared as optional, then still default
value is given by Request param is null only. This we can not store
in variable. That gives Http Status -500 - INTERNAL SERVER ERROR.

*) Controller

```
package in.nareshit.raghu.controller;
```

```
import java.util.List;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```

@Controller
public class StudentController {

    //.... /show?sid=10&sname=ABCD

    @GetMapping("/show")
    public String readData(
        @RequestParam(value = "sid", required = false)
int id,
        @RequestParam(value = "sname", required =
false, defaultValue = "NONE")String name,
        Model model
    )
    {
        model.addAttribute("id", id);
        model.addAttribute("name", name);
        return "StdHome";
    }
}

```

```

*) UI : StdHome.jsp
<%@ page isELIgnored="false" %>
<html>
<body>
Data is : ${id}, ${name}
</body>
</html>

```

Req URL:
http://localhost:9898/show?sname=ABCD
Output:
type=Internal Server Error, status=500

At Console:
IllegalStateException:
Optional int parameter 'sid' is present but
cannot be translated into a null value

due to being declared as a primitive type.

Consider declaring it as object wrapper for the
corresponding primitive type.

Sol#1 (Given by Spring WEB MVC)

Convert Primitive to Wrapper

```

[Old code]
@RequestParam(value = "sid", required = false) int id

{new code}
@RequestParam(value = "sid", required = false) Integer id

```

Sol#2 (Given by Raghu Sir)

Modify Default value from null to any other

[old code]

```
@RequestParam(value = "sid", required = false) int id
```

[new code]

```
@RequestParam(value = "sid", required = false, defaultValue="-1") int  
id
```
