

what is the difference designing form pages using html form tags and spring mvc supplied jsp form tags?

Traditional html form tags

(a) These tags are given by w3c

(world wide web consortium)

(b) These tags support one way data binding when they are used 'spring mvc apps (form data to Model class obj)

(c) default request method for the form page

is GET

are

(d) These tags not strictly typed

are

(e) These tags not case-sensitive

(f) can be used in html programming and

also in jsp programming (.html, .jsp files)

spring MVC supplied jsp form tags -by-----

(a) These tags are given spring MVC framework (pivotal team)

b) These tags support two way data binding when they are used spring mvc apps

(form data to Model class obj and vice-versa)

(c) default request method for the form page is POST

are

(d) these tags strictly typed

(e) These tags are very much case-sensitive

(f) can be used only in jsp programming (only in .jsp files)

Two different ways of data binding

Data binding using @RequestParam

(Binding request param values to handler method params

)

1. Form Data Binding

to Model class obj using @ModelAttribute Annotation

2. request param values

binding to handler method

params using @RequestParam annotation

=> sometimes hyperlinks carry additional data along with the generated request in the form of request params which will be appended to the request url as part of query String

(href url)

 edit student

 edit student

values

url having query String (request params)

These two hyperlinks

generate GET mode request data carrying additional as the value of "sno" request param

=>These request params can be bound with Handler method params using the support of @RequestParam annotation

request path "

ex1:: request url::

http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=1001&sname=rajesh

In controller class

=====

Values

@GetMapping("/data")

request params are bound to handler method params

```
public String process(@RequestParam("sno") int no, @RequestParam("sname") String name) {
```

```
System.out.println(no+" "+name);
```

```
//return LVN
```

```
return "show_result";
```

```
}
```

syntax of @RequestParam

(Needs to do this if req param name

@RequestParam("param name") data type param name (or)

and method param name are not matching)

@RequestParam data type param name (Needs to do this if the request param name and method param name are matching)

=> if request param names are matching with handler method param name then there is no need of giving request param names in @RequestParam annotation

ex2 :

http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=9001&sname=rakesh

Handler method in controller class

@GetMapping("/data")

I

```
public String process(@RequestParam int sno, @RequestParam String sname) { System.out.println(sno+" "+sname);
```

```
}
```

```
//return LVN
```

```
return "show_result";
```

If any request param name is not matching with @RequestParam attribute name or method param name of

handler method then we get exception

[org.springframework.web.bind.MissingServletRequestParameterException: Required request parameter 'sname' for method

parameter type String is not present]

ex3::

http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=6001&name=mahesh

Handler method of controller class

=====

@GetMapping("/data")

matching

not matching

so binding takes place

sol400 error status will be raised.

public String process(@RequestParam int sno, @RequestParam String sname) { System.out.println(sno+" "+sname);

throws Exception

//return LVN

}

return "show_result";

By default @RequestParam based method param value should be filledup becoz the default value for required attribute in @RequestParam is true.

Solution for ex3 problem

(take required=false in @RequestParam annotation) optional to place

(To avoid the exception though request param name is not matching with method param name)

request url :: http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=1021&name=raju

solution2:

Handler method of controller class

@GetMapping("/data")

matching

not matching

so null will be taken

public String process(@RequestParam int sno, @RequestParam(required = false) String sname) { System.out.println(sno+" "+sname); gives 1021 null

}

//return LVN

return "show_result";

(take default value for the request param in @RequestParam)

http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=1022&name=rajesh

<http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=1022>

<http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=1022&sname=king>

handler method of controller class

```
@GetMapping("/data")
```

```
public String process(@RequestParam int sno,
```

```
}
```

```
@RequestParam(defaultValue = "rrr") String sname) { System.out.println(sno+" "+sname); //return LVN
```

```
1022 rrr
```

```
return "show_result";
```

```
1022 rrr 1022 king
```

What the difference b/w required="false" and defaultValue of @RequestParam annotation?

=> required=false says though certain req param value is not given in the request url then it should not give any exception rather assigns null value method param (Default value to method param)

=> defaultValue="xxxx" assigns the given default value method param if the req param is

or request param name is not matching with method param name

given

For numeric type method param if we try to pass string type value from request param then we get Number Format Exception.. No Solution for this problem except passing correct format value.

case1:: <http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=101f&sname=raja>

<http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=101f&sname=raja>

In controller class

```
@GetMapping("/data")
```

```
public String process(@RequestParam("sno") int no,
```

```
throws exception
```

```
@RequestParam("sname") String name) {
```

```
System.out.println(no+" "+name);
```

```
//return LVN
```

```
}
```

```
return "show_result";
```

raises :: 2022-03-18 19:37:46.115 WARN 6872 --- [nio-2020-exec-5] .w.s.m.s.DefaultHandlerExceptionResolver: Resolved [org.springframework.web.method.annotation.MethodArgumentTypeMismatchException: Failed to convert value of type 'java.lang.String' to required type 'int'; nested exception is java.lang.NumberFormatException: For input string: "101f"]

=> for number type method param of handler method,if u try to pass no value or empty string' then

also we get NumberFormatException.. This problem can solved using "defaultValue" or required false with

Wrapper type params.

[http://localhost:2020/BootMVCProj06-DataBinding-RequestParam Binding/data?sno=&sname=raja](http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data?sno=&sname=raja)

In controller class

```
@GetMapping("/data")
```

```

public String process(@RequestParam("sno") int no, @RequestParam("sname") String name) {
    System.out.println(no+" "+name);

    //return LVN
    return "show_result";
}

```

raises :: 2022-03-18 19:38:36.517 WARN 6872 --- [io-2020-exec-11] .w.s.m.s.DefaultHandlerExceptionResolver: Resolved [org.springframework.web.method.annotation.MethodArgumentTypeMismatchException: Failed to convert value of type 'java.lang.String' to required type 'int'; nested exception is java.lang.NumberFormatException: For input string: ""] for

Soultion1: Assign default value handler method parameter

http://localhost:2020/BootMVCProj06-DataBinding-RequestParam Binding/data?sno=&sname=raja

Handler method of controller class

```
@GetMapping("/data")
```

```

public String process( @RequestParam(name="sno",defaultVlaue="2001") int no,
}

```

```
@RequestParam("sname") String name) {
```

gives 2001 raja

```
System.out.println(no+" "+name);
```

```
//return LVN
```

```
return "show_result";
```

solution2: take required=false having wrapper type method param

http://localhost:2020/BootMVCProj06-DataBinding-RequestParam Binding/data?sno=&sname=raja

Handler method in controller class

```
@GetMapping("/data")
```

```

public String process(@RequestParam(name = "sno",required = false) Integer no, @RequestParam("sname")
String name) { gives null raja

```

```
System.out.println(no+" "+name);
```

```
//return LVN
```

```
}
```

```
return "show_result";
```

=>if we pass same request param with multiple values in the query String to bind them for handler method parameter we can the parameter type as the array or List or Set type as shown below

request url:: http://localhost:2020/BootMVCProj06-DataBinding-RequestParamBinding/data

In the handler method controller class

?sno=1022&sname=raja&sadd=hyd&sadd=vizag&sadd=delhi

3 request params with same name and diff values

```
@GetMapping("/data")
```

```
public String process(@RequestParam(name = "sno",required = false) Integer no,
```

```
@RequestParam("sname") String name, @RequestParam("sadd") String addrs[], @RequestParam("sadd")
```

```
List<String> saddList, @RequestParam("sadd") Set<String> saddSet
```

```
System.out.println(no+" "+name+" "+Arrays.toString(addr)+ " "+saddList+" "+saddSet); //return LVN gives  
1022 raja [hyd, vizag, delhi] [hyd, vizag, delhi] [hyd, vizag, delhi]
```

```
return "show_result";
```

```
}
```

if the handler method param type is simple string and we are trying to give multiple values through request param

the simple string param holds the multiple values as the comma separated list of values.

request url ::

*http://localhost:2020/BootMVCProj06-DataBinding-RequestParam
Binding/data?sno=101&sname=raja&sadd=hyd&sadd=vizag*

handler method of controller class

```
@GetMapping("/data")
```

```
public String process(@RequestParam(name = "sno",required = false) Integer no,
```

```
System.out.println(no+" "+name+" "+addr);
```

```
@RequestParam("sname") String name,
```

```
@RequestParam("sadd") String addr) {
```

```
//return LVN
```

```
101 raja hyd,vizag
```

```
return "show_result";
```

```
}
```

note:: if form page is having group of checkboxes or List box which allows to select multiple items at a time then we need to take

array or List of Set type properties in Model class that will be used in form data binding.

Use case of @RequestParam in real applications

use case 1::

webpage

=====

```
@Controller
```

```
generateExcel report
```

```
href= report?type=excel
```

```
generate pdf report href=report?type=pdf]
```

```
public class ReportGenerationController{ @Autowired
```

```
private ReportDataService service;
```

```
@GetMapping("/report")
```

```
public String generateReport(Map<String,Object> map,
```

```
@RequestParam(name="type") String type){
```

```
List<Product> list-service.getProductsInfo();
```

```

map.put("reportData",list);
if(type.equalsIgnoreCase("excel")){
    return "excel report"; //LVN
}
else if(type.equalsIgnoreCase("pdf")){
    return "pdf_report"; //LVN
}
}
}
}

```

Here we are differentiating logics for multiple hyperlinks generated requests in the handler method based the additional request param (type) value sent by those hyperlinks

UseCase2:

ReportPage (jsp page)

PID PNAME PRICE QTY operations

101 table 90.44 10 edit delete

[edit?o=101 delete?no=101](#)

102 chair 45.77 19 edit delete

[edit?o=102 delete?no=102](#)

@Controller

```

public class CURDOperationsContrller{
    @GetMapping("/edit")
    public String editProduct(Map<String,Object> map,
    ...
    @RequestParam("no") int no){
        //invoke service logic to modify the record
    }
    @GetMapping("/delete")
    public String deleteProduct(Map<String, Object> map,
    @RequestParam("no") int no){
        //invoke service logic to delete the record
    } }//class

```

if handler method of controller class is expecting

more than that.. then we will not get any problem.

only few req param from the reuquest.. but if we give

note::

<http://localhost:2525/MVCBootProj06-DataBindingUsingRequestParams/data?sno=1002&sname=raja&sadd=hyd&avg=90.0&school=HPS>

@GetMapping("/data")

```
public String process(@RequestParam Integer sno, @RequestParam String sname, @RequestParam("sadd")
String addr) { System.out.println(sno+" "+sname+" "+addr);
}
```

//return LVN

```
return "show_result";
```

gives 1002

raja hyd

if we pass more than or equal to expected request param values

then no problem .. if we pass less the expected request params then

there is possibility of getting errors.. So solve this error take required=false or/and defaultValue="xxx" in
@RequestParam annotation

coming

in the handreams for which vakues are not as the request params

additional and unwanted request params.. but they

do not create any problem

MVCBootProj06-DataBinding Using RequestParams [boot]

>

Deployment Descriptor: MVCBootProj06-DataBinding Usin

> Spring Elements

JAX-WS Web Services

#src/main/java

✓ com.nt

> MvcBootProj06DataBinding UsingRequestParamsAp

com.nt.controller

>

ServletInitializer.java

>RequestParamsController.java

src/main/resources

static

templates

application.properties

>src/test/java

> JRE System Library [JavaSE-16]

> Maven Dependencies

> L Deployed Resources

✓ src

main

> java

>

resources

✓

webapp

✓

WEB-INF

pages

show_result.jsp

> test

=>Using **@Autowired**, we can inject **ServletContext** obj,**ServletConfig** obj to the controller class =>We can get **HttpSession** obj, **HttpServletRequest**, **HttpServletResponse** objs as the params of handler method in controller class

In **@Controller** class

=====

@Autowired

private ServletContext sc;

@Autowired

private ServletConfig cg;

@GetMapping("/data")

public String showData(HttpSession ses, HttpServletRequest req,HttpServletResponse res) throws Exception{

System.out.println("web app name::"+sc.getContextPath());

System.out.println("DS logical name::"+cg.getServletName());

System.out.println("Session id ::"+ses.getId());

System.out.println("current req uri::"+req.getRequestURI());

System.out.println("response content type::"+res.getContentType());

return "welcome";

}

=>**ServletContext** object in handler method is useful to get more information about the underlying web application =>**ServletConfig**.object in handler method is useful to get more information about **DispatcherServlet** =>**Session** object" handler method is useful for Session Tracking

=> **HttpServletRequest** object in handler method is useful to gather more details bout incoming request

=> **HttpServletResponse** object in handler method is useful to set more details bout outgoing response and also useful to send response to browser directly from handler method with out taking support of **ViewResolver** and

View comps

@GetMapping("/data")

public void processData(Map<String, Object> map,HttpServletRequest req, HttpServletResponse res, HttpSession ses) throws Exception{

```
// get PrintWriter object
```

```
PrintWriter pw=res.getWriter();
```

```
//set response content type
```

```
res.setContentType("text/html");
```

```
//write the output to browser
```

```
pw.println("request Path:"+req.getServletPath());
```

This handler method sends the generated output directly to browser

with out involving ViewResolver and view comps i.e useful in FileDownloading

or Software Downloading activities

0