



Understanding what is returned (5th class)

Presented by ARMS (THAILAND) Co., Ltd.

Index

1. Sample project

- 1-1. Summary
- 1-2. RequestParam
- 1-3. Returning JSON
- 1-4. PathVariable
- 1-5. ModelAttribute

1. Sample Project

1-1. Summary

- It is significantly important to understand what values are returned. whether it returns path, parameter, JSON, or View?

1. Sample project

1-2. RequestParam

- Create a controller and run this project.

```
package com.arms;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {

    @RequestMapping("/list")
    public String index(@RequestParam("name") String name){

        return "Hello, " + name;

    }

}
```

Access

<http://localhost:8080/list?name=John>

1. Sample project

1-2. RequestParam

- Change from `@RestController` to `@Controller` and run this project

```
package com.arms;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@Controller
public class HelloController {

    @RequestMapping("/list")
    public String index(@RequestParam("name") String name){

        return "Hello, " + name;

    }

}
```

Access

<http://localhost:8080/list?name=John>

Error occurs....?

1. Sample project

1-2. RequestParam

- `@Controller` returns the View while `@RestController` returns response contents. Add `@ResponseBody` annotation to return contents.

```
package com.arms;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class HelloController {

    @RequestMapping("/list")
    @ResponseBody
    public String index(@RequestParam("name") String name){
        return "Hello, " + name;
    }
}
```

`@Controller + @ResponseBody = @RestController???`

1. Sample project

1-3. returning JSON

- Create a Controller to return JSON

```
package com.arms;

import java.util.Collections;
import java.util.Map;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class HelloJsonController {

    @RequestMapping("/json")
    @ResponseBody
    public Map<String, String> index(){
        return Collections.singletonMap("welcome", "Hello Springboot");
    }

    @RequestMapping("/")
    public String getJn(){
        return "index";
    }
}
```

Access <http://localhost:8080/json>

You'll see JSON format data

1. Sample project

1-3. returning JSON

- Use JQuery ajax to receive JSON type data.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Ajax to receive JSON</title>
</head>

<body>
<div>
    <span class="label">Welcome Message : </span>
    <span class="value"></span>
</div>
</body>
<script src="https://code.jquery.com/jquery-1.12.1.min.js"></script>
<script type="text/javascript">
$(function() {
    $.ajax({
        url: '/json',
        method: 'get',
        cache: false
    }).then(function(data) {
        $('<span class="value">').text(data.welcome);
    });
});
</script>
</html>
```

Ajax option parameter

url : the URL to which the request is sent
cache: set this to false to force the page not to cache by browser.

You'll see "welcome" key to get
"Hello Springboot" value

Access <http://localhost:8080/>

1. Sample project

1-4. PathVariable

- Create a project and a controller as below to understand PathVariable

```
package com.arms;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class HelloController {

    @RequestMapping("/list/{id}/{name}")
    @ResponseBody
    public String index(@PathVariable("id") int id, @PathVariable("name") String name){
        return "UserName: " + name + " ID: " + id;
    }
}
```

Access <http://localhost:8080/list/7/Mike>
And see PathVariable get URI values

1. Sample project

1-4. PathVariable

- PathVariable is used to get RESTFUL URI

`list/7`

- RequestParam is used to get a parameter as the name suggests.

`list/?id=7`

`list/?name=John`

1. Sample project

1-5. ModelAttribute

- ModelAttribute is used when clients send data by **x-www-form-urlencoded**.
- **x-www-form-urlencoded** is used when clients send form data to web server (Content-Type name)

x-www-form-urlencoded data pattern: id = data

If there are multiple form parameters, the following pattern is used
id=data1&form2=data2&form3=data3 (& is delimiter)

Data is URL encoded

_ . – are used as it is, space is +, other characters are %xx hex(16)

1. Sample project

1-5. ModelAttribute

- Let's see how @ModelAttribute works

```
package com.arms;

public class Greeting {

    private int id;
    private String content;

    public int getId(){
        return id;
    }

    public void setId(int id){
        this.id = id;
    }

    public String getContent(){
        return content;
    }

    public void setContent(String content){
        this.content = content;
    }

}
```

Create Greeting.java

1. Sample project

1-5. ModelAttribute

- Let's see how @ModelAttribute works

Create GreetingController.java

```
package com.arms;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
public class GreetingController {

    @RequestMapping(value = "/greeting", method=RequestMethod.GET)
    public String GreetingForm(Model model){
        model.addAttribute("greeting", new Greeting());
        return "greeting";
    }

    @RequestMapping(value = "/greeting", method=RequestMethod.POST)
    public String greetingSubmit(@ModelAttribute Greeting greeting){
        return "result";
    }
}
```

1. Sample project

1-5. ModelAttribute

- Let's see how @ModelAttribute works

Create greeting.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Form Input</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
  <h1>Greeting Form</h1>
  <form action="#" th:action="@{/greeting}" th:object="${greeting}" method="post">
    <p>Id: <input type="text" th:field="**{id}" /></p>
    <p>Message: <input type="text" th:field="**{content}" /></p>
    <p><input type="submit" value="Submit" /> <input type="reset" value="Reset" /></p>
  </form>
</body>
</html>
```

1. Sample project

1-5. ModelAttribute

- Let's see how @ModelAttribute works

Create result.html

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>FormResult</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
  <h1>Result</h1>
  <p th:text="'id: ' + ${greeting.id}" />
  <p th:text="'content: ' + ${greeting.content}" />
  <a href="/greeting">Submit another message</a>
</body>
</html>
```

After this, run your project and access
localhost:8080/greeting

1. Sample project

1-5. ModelAttribute

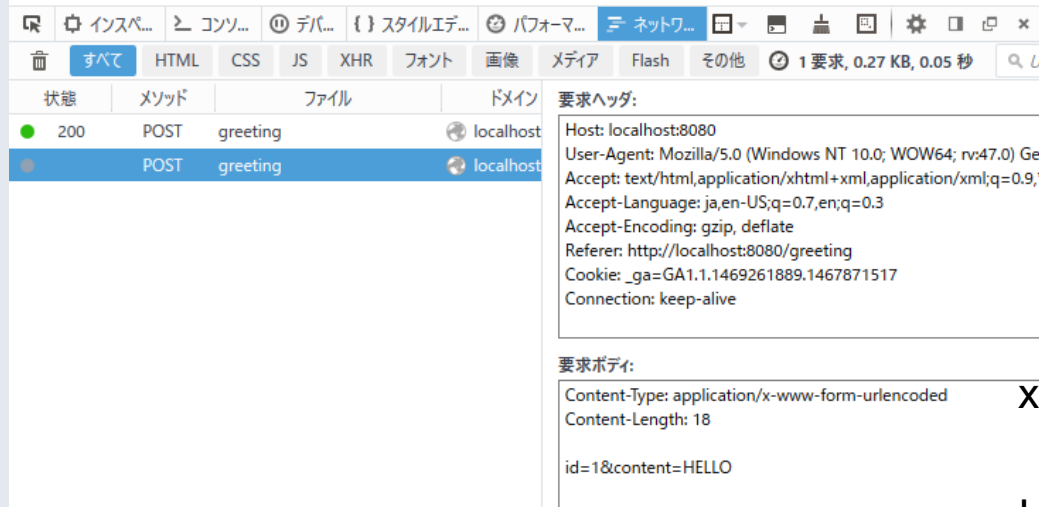
- Let's try to submit form data and see what happens.

Result

id: 1

content: HELLO

[Submit another message](#)



x-www-form-urlencoded

Id=1&content=HELLO

1. Sample project

1-5. ModelAttribute

- What if you change @ModelAttribute to @RequestBody.

GreetingController.java

```
@RequestMapping(value = "/greeting", method=RequestMethod.POST)
public String greetingSubmit(@RequestBody Greeting greeting){
    return "result";
}
```

x-www-form-urlencoded can not receive

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Aug 30 13:42:07 ICT 2016

There was an unexpected error (type=Unsupported Media Type, status=415).

Content type 'application/x-www-form-urlencoded;charset=UTF-8' not supported



Your Idea Leads Your Ideals

homepage: <http://arms-asia.com/>

facebook: <https://www.facebook.com/arms.asia?fref=ts>