

## Making Spring Rest Producer App Sending XML Response body to Client App Consumer App

(@RestController)

=> Based on given request type, making producer App sending different type of content with the specified content type is technically called Content negotiation.

=> For this we need to use "Accept" request header specifying content type like "JSON", "XML" and etc.. This makes producer application giving content in response in the requested format .. Through 'Accept' is request header, it makes the Producer/server/service provider App sending

its content in required format like JSON, XML and etc..

=> By default every method in Producer App sends plain text response if the return type is

ResponseEntity<String> otherwise (for ResponseEntity<non-String> it sends JSON response) sends JSON format content

Making Producer Application Sending XML response

=====

=====

=> For this the Client app/consumer App must set "Accept" request header to "application/xml" (default is /\*)

Accept: application/xml tells to producer App give me response content as xml content

Accept: /\* tells to Producer Apps give me response in any format (what u want, that u can give) => We need to add the following additional dependency (jar file) to the project for converting Object data to xml format data

<!-- [https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-](https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-dataformat-xml)

[dataformat-xml](https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-dataformat-xml) -->

<dependency>

<groupId>com.fasterxml.jackson.dataformat</groupId> <artifactId>jackson-dataformat-xml</artifactId>

<version>2.12.5</version>

</dependency>

Consumer App side

req header Accept :: /\*

(default)

req header accept:: /\* (or) application/JSON

req header accept :: application/XML

collect from mvnrepository.com

non-string ::

Producer App side

Response format of producer App

text content

# 三權

Rest operation method's return type `ResponseEntity<T>` generic type is `<String>` Rest operation method's return type `ResponseEntity<T>` generic type is `<non-string>` Rest operation method's return type `ResponseEntity<T>` generic type is `<non-string>`

JSON Content

XML Content

Model class or collection or array or collection of Model class objs

Client App (Consumer App)

Http Request

GET

Accept: application/xml

HEAD

(request header)

Producer App

`@RestController`

`@GetMapping("</path>")`

`public RT <method>(){ (other than String)`

....

`return object;`

Jackson-dataformat-xml must be added to CLASSPATH

Http Response

}

http 1.1

200 OK

HEAD

object to xml

xml content

as the

response

`<>.....`

`</>` (xml body)

Body

Example App

step1) create Spring Boot starter project having

following starters, dependencies (spring web, lombok, jackson-dataformat-xml)

step2) Develop the Model class

//Model class

**@Data**

**@NoArgsConstructor @AllArgsConstructor**

public class Customer {

private Integer cno;

private String cname;

private String cadd;

private Float billamt;

(starters)

(dependencies)

}

step3) Develop Producer as @RestController having method with other String Return type.

//Producer App

package com.nt.controller;

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

org.springframework.web.bind.annotation.RestController;

import com.nt.model.Customer;

**@RestController**

public class CustomerOperationsController {

**@GetMapping("/report")**

public Customer showData() {

Customer cust=new Customer(1001,"raja","hyd",44545.77f); return cust;

(or)

}

step4) Run the Application

(RunAs --- Run on server)

*Default response status code 200 will be taken*

**@GetMapping("/report")**

public ResponseEntity<Customer> showData() {

Customer cust=new Customer(1001,"raja","hyd",44545.77f); return new

ResponseEntity<Customer>(cust,HttpStatus.OK);

*Taking ResponseEntity<T> as Return is recommended because it gives control on response status code.*

}

step5) Give Request from POSTMAN Tool by setting "Accept" header with the value "application/xml"

(a)

GET

http://localhost:3030/Spring Boot RestProj04-XML Response/report (b)

Params Authorization Headers (7) Body

Send

(f)

Host

(c)

Pre-request Script Tests Settings <calculated when request is sent>

Cookies

✓ User-Agent

✓

Accept (deslect) (d)

Accept-Encoding

< Connection

PostmanRuntime/7.28.4

\*/\*

gzip, deflate, br

keep-alive

Accept

(e)

application/xml

Response

Body Cookies Headers (5)

Test Results

Pretty Raw

Preview

Visualize

XML

1

<Customer>

2

<cno>1001</cno>

(g)

3

<cname>raja</cname>

4

<cadd>hyd</cadd>

5

```
<billamt>44545.77</billamt>
```

6

```
</Customer>
```

200 OK 292 ms 275 B

[Activate Wir](#)

[Go to Settings t](#)

**=>if we send the above request with out placing Accept: application/xml request header then Accept:\*/\* will be taken**

**as default value and we get Json response as show below**

Body Cookies Headers (5) Test Results

200 OK 15 ms 223 B

Pretty Raw

Preview

Visualize

JSON

=

1

2

```
"cno": 1001,
```

3

```
"cname": "raja",
```

4

```
"cadd": "hyd",
```

5

```
"billamt": 44545.77
```

[Activate Wir](#)

[Go to Settings to](#)

**=>In the above setup where we asking for XML response using "Accept:application/xml" request header, if we are not placing "jackson-dataformat-xml" then we get 406 (Media Type Not Acceptable) Error**

406 Not Acceptable 127 ms

**=> we run the same Application with the above setup (not adding jackson-dataformat-xml jar file) and with out chaging "Accept" header value to "applicaiton/xml" then we get JSON response fron producer App becoz the method return non-String type**

Body Cookies Headers (5) Test Results

Pretty

Raw

Preview

1

2

"cno": 1001,

3

"cname": "raja",

4

"cadd": "hyd",

5

"billamt": 44545.77

6

Visualize

JSON

-+

200 OK

Av

Gc

**Q) Can we take return type of the rest operation method with out ResponseEntity? Ans) possible, if the return typeString then we get text response and if the return type is other than String then we get either XML or JSON format response based on the "accept" req header value we send from the consumer app.**

**note:: Spring Rest Apps /Rest APIs/spring Rest provider apps**

**themselves do not present data using UI comps, the take inputs in the form of XML or JSON from the Rest API Provider App and sends output to front end app in the form of XML or JSON content (consumer app)**

**In @RestController class**

**@GetMapping("/cust\_report")**

**public Customer reportCustomerData(){**

**This Rest operation method**

**returned response status code**

**always 200 and it can not**

**Customer cust=new Customer(1001,"raja", "hyd",90000.0); be changed.. return cust;**

**if we take the Return type as**

**}**

**To get better control on response status code and response headers while returning results from the endpoints of**

**the RestController/Provider App its good to take Response Entity<T>**

**the ResponseEntity<T> then we can as the return type of the end point method send result + response status code together as the response.**

**note:: overall ResponseEntity<T> as the return type is always good (came along with the request) Working with @RequestBody to convert given XML/JSON content into Model class obj**

**request**

=> if Client generated http request is having XML/JSON content as the body and if want to construct java class object having that data then we need to specify `@RequestBody` for the parameter of Producer method whose type (endpoint is Model class name/Java class name

Client App/Consumer

App

POST

HEAD

...

`{k:v,k:v,...}` (JSON body Body

Producer/Server App

=>GET, HEAD mode request does not contain

body part.. but they send little amount

of data as request param names and values in query String eg: `url?param1=val1&param2=val2`

=> The response of GET mode request contains both HEAD, BODY part where the response of HEAD mode request does not contains BODY Oplect for class part.. So we use GET mode request to get Here JSON data from server and HEAD mode request data will be converted obj) to check wheather certain web comp is avaible or not (Debugging purpose)

(we can not to use GET mode here `@PostMapping("</path>")` becoz GET mode reuest does public `<RT>` `<method>(@RequestBody contain body)`

to

`<Classname> param){`

=>GET mode request carry the data

in the form of query string appended

to the url, so GET mode request does

not BODY in the request structure contain

=> POST mode request carry the data in the form of request body content i.e data will not be appended to the url as the query string.

}

(while working with this Application `jackson-dataformatter-xml` is optional)

`@RequestBody` and `@ResponseBody` annotations are called MediaType annotations becoz they to recognize the input,output types like JSON content,XML content and etc..

`@RequestBody` is useful to convert the request related JSON/XML content to JAVa class object

Rules with respect

to Deserilization activity

(Converting JSON doc content to

`@ResponseBody` is useful to convert rest operation method generated other than String

Model class object content) content to JSON/XML content

we do not use `@ResponseBody` separately, rather it will be applied automatically

on every rest operation method through `@RestController` kept on the class name

Syntax to use `@RequestBody` in method param

`@RequestBody <ClassName> <paramName>`

(no keys, values)

becomes obj name internally

=> if JSON body is coming as {} (Empty folower bracket) then the Object for class given in `@RequestBody` will be ceated using O-param constructor with default values giverWM (0/0.0/null/false)

(Model class)

=> The keys in JSON body and the Property names in the giverclass of `@RequestBody` must match

Example App

in order to see Data binding to object.. if few keys and property names are matching and others not matching then binding will happen only for few properties the remaining properties hold default values.

=====

step1) Create spring Boot Project

adding the following starters, dependendencies

a) Lombok b) spring web c) jackson-dataformat-xml

(required for giving

step2) Develop the Model class xml content in request `@RequestBody` of spring Rest is very much similar to

//Model class

```
package com.nt.model;
```

```
body)
```

`@ModelAttribute(-)` of spring MVC

`@ModelAttribute(-)` is useful to bind IN MVC module

form data into Model class object `@RequestBody` is useful to bind JSON/XML data of request into Model class object

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

`@Data`

`@NoArgsConstructor`

```
@AllArgsConstructor public class Customer { private Integer cno; private String cname; private String cadd;
private Float billAmt;
```

note :: we need to implement

java.io.Serializable interface

on Model class only if there

is a need of sending that Model

class object data over the network

```
}
```



### step3)

Develop Producer App as `@RestController`

or need of writing Model class obj data

to a file using Serialization concepts

note: if want to use Model class obj to carry data between two layers of same Project then there is no need of implementing `java.io.Serializable` on it.

```
package com.nt.controller;
```

```
import org.springframework.web.bind.annotation.PostMapping; import
org.springframework.web.bind.annotation.RequestBody; import
org.springframework.web.bind.annotation.RestController; import com.nt.model.Customer;
```

`@RestController`

```
public class CustomerOperationsController {
```

`@PostMapping("/register")`

```
public String registerCustomer(@RequestBody Customer cust) {
```

```
    return cust.toString();
```

```
}
```

```
}
```

### step4) Run the Application

### step5) Send the following request from POSTMAN

(a)

POST

`http://localhost:3030/Spring Boot Rest Proj05-JSON-XMLRequestBody/register` (b)

Params

Authorization Headers (8)

Body 

Pre-request Script Tests Settings (c)

none

 form-data

 x-www-form-urlencoded

 raw  binary  GraphQL

JSON

(d)

1

```
{"cno":1001,"cname":"raja", "cadd": "hyd", "billAmt":5679.0}
```

type json body (e)

Send (f)

c

In Rest module

```
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-dataformat-xml --> <dependency>
<groupId>com.fasterxml.jackson.dataformat</groupId> <artifactId>jackson-dataformat-xml</artifactId>
</dependency>
```

**Version is not required for jackson jar file**

200 OK 11 ms 220 B

Body Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text

1

Customer (cno=1001, cname-raja, cadd-hyd, billAmt=5679.0)

**(g) observe this output**

**send**

=> After taking request body as JSON/XML and if we do not correct format body content along with the `HttpRequest` then we get 400 error response (Bad Request)

(no keys, values)

=> if JSON body is coming as {} (Empty folower bracket) then the Object for class given in `@RequestBody` will be ceated using O-param constructor with default values

giver WM (0/0.0/null/false)

if the model class is having `java.time.LocalDateTime` property then we can give value to that

**property in json doc in this pattern yyyy-MM-ddTHH:mm:ss**

**example**

In Model class

**private LocalDateTime expiry;**

**NTSPBMS616 / test5-req2 o**

POST

A Save

`http://localhost:4040/BootRestProj05-JSON-XMLDataBindingToObjectUsing`  
`RequestBodyAnnotation/productapi/registre`

Params

Authorization

Headers (8)

Body ●

none form-data

☒ x-www-form-urlencoded

raw

Pre-request Script ☐ binary

Tests Settings

• GraphQL

JSON

1

2

3

Body Cookies Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text ✓

1

Send

In JSON document

"expiry":"2023-11-23T17:25:34"

POST

目

http://localhost:3131/Boot Rest Proj05-ProviderApp-RequestBody-JSON/customer-api/register

Params Authorization Headers (8) Body ☒ Pre-request Script Tests Settings

Cookies

none

form-data

Ox-www-form-urlencoded

Beautify

1 {

2

Meeting Chat

3

- "cno": 101,

no error

4

Status: 201 Created Time: 5 ms Size: 247 B

```
"cname": "raja",
```

Murali Krishna to You (

5

```
"cadd": "hyd",
```

yes sir clear

6

```
"billAmt":987878,
```

Mohit Chotalia to You (

0

7

```
-"doBlling": "2024-03-10T10:11:34"
```

sir tomorrow

class

8

9

}

Product (pid=null, pname=null, price=null, qty=null, expiry=null) is registered

**Sending Xml data as the request body along with**

**using @RequestBody**

===

**Client App/Consumer POST**

...

App

**HEAD**

<>...</> (xml body)

(we can not to use GET mode here becoz GET mode reuest does  
contain body)

=====

**Producer/Server App object for**

class

Send

Cookies

O raw

O binary GraphQL JSON

Beautify

ody Cookies

Headers (5)

Test Results

Pretty Raw

Preview

Visualize

Text

<

(Here Xml data will be converted obj)

`@PostMapping("</path>") public <RT> <method>(@RequestBody`

(while working with

this Application

jackson-dataformatter-xml is

`<Classname> param){`

mandatory)

Example App (same as above)

but make sure that Jackson data format -xml jar file is added to the build path

Test the App using Post Main as show below

POST (a) `http://localhost:3030/SpringBoot RestProj05-JSON-XMLRequestBody/register` (b) Authorization Headers  
(8) Body ●

Params

none ● form-data ● x-www-form-urlencoded

Pre-request Script Tests Settings

(c)

● raw ● binary

● GraphQL XML

(d)

`<eno>456</eno>`

`<cname>rajesh</cname>`

1

`<customer>`

2

3

4

5

6

`</customer>`

`<cadd>vizag</cadd>`

<billAmt>5678</billAmt>

**(e) type this content**

Body Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

Text

1

Customer (cno=456, cname=rajesh, cadd=vizag, billAmt=5678.0)

**Flow of execution**

=====

POST

O 200 OK 22 ms 251 B

Save as example

000

1 Customer (cno=101, cname=raja, cadd=hyd, billAmt=987878.0,  
doBlling=2024-03-10T10:11:34)

POST

http://localhost:3131/Boot Rest Proj06-ProviderApp-RequestBody-XML/customer-api/register

**Send (f)**

Params Authorization

Headers (8)

Body ●

none

O form-data

Ox-www-form-urlencoded

Pre-request Script

Tests

Settings

XML

O raw O binary O GraphQL

1

<customer>

2

</customer>...

Headers (5) Test Results

Raw

Preview

Visualize

Text

Body

Cookies

200 OK 6 ms

Pretty

**(g) get this output**

Go to Se

http://localhost:3030/SpringBootRestProj05-JSON-XMLRequestBody/register

Params

Authorization

Headers (8) Body ☒ Pre-request Script Tests Settings

☒ none

x-www-form-urlencoded

☒ raw binary

GraphQL XML

1 <customer>

2

<cno>456</cno>

3

<cname>rajesh</cname>

4

<cadd>vizag</cadd>

5 <billAmt>5678</billAmt>

6 </customer>

Body

Cookies

Headers (5) Test Results

**(Response)**

Pretty

Raw Preview

Visualize

Text

⌘

(1)

**send**

(http request

=

200 OK

21 ms

234 B

```
1 Customer (cno=null, cname=null, cadd=null, billAmt=null, doBlling=null)
```

**with Body generated)**

(3)

**HandlerMapping**

**on**

(2) (11)

**DispatcherServlet**

(6) gets details from

**HandlerMapping**

(7) Based **@RequestBody**

param, it uses different api

(5) gets **registerCustomer(-)**

signature **+@RestController** bean id (CustomerOperationsController)

(8) invokes the

**the method of controller having Customer**

**to convert xml content of**

**obj as the arg value**

request to customer class obi data

**Customer obj(cust)**

```
1 Customer(cno=456, cname=rajesh, cadd=vizag, billAmt=5678.0) (12)
```

**@RestController**

```
public class CustomerOperationsController {
```

```
    @PostMapping("/register")
```

(9)

(4?) search for Post mapping method with /register as request path +POST mode

```
    public String registerCustomer(@RequestBody Customer cust) { return cust.toString(); (10)
```

```
}
```

=> if request body content type is XML and the producer is not having

jackson-dataformat-xml jar file in the build path then we get



**415 error**

(415 Unsupported Media Type)

**conclusion**

**to get**

**(request header)**

**406 :: we want Xml response through "Accept:application/xml" but jackson-dataformatter-xml jar is Problem towards**

**(Not acceptable not added in the producer application (related @ResponseBody))**

**reflects the problem related**

**marshalling activity**

**to data rendering**

**415 ::**

**we want to send xml body along with request to store into Object using @RequestBody**

**Problem towards**

**Reflects the problem related to data binding**

**(unsupported Media)**

**but jackson-dataformatter-xml jar is not added in the producer application**

**unmarshalling activity**

**Send**

**Cookies**

**Beautify**

**A Save as example**

**000**

**8**