

Spring Framework

28. Spring RESTful 웹서비스 어플리케이션 작성(4)

CONTENTS

1

XML 응답을 주는 RESTful 웹서비스 개요

2

XML 응답을 주는 RESTful 웹서비스 구현

학습 목표

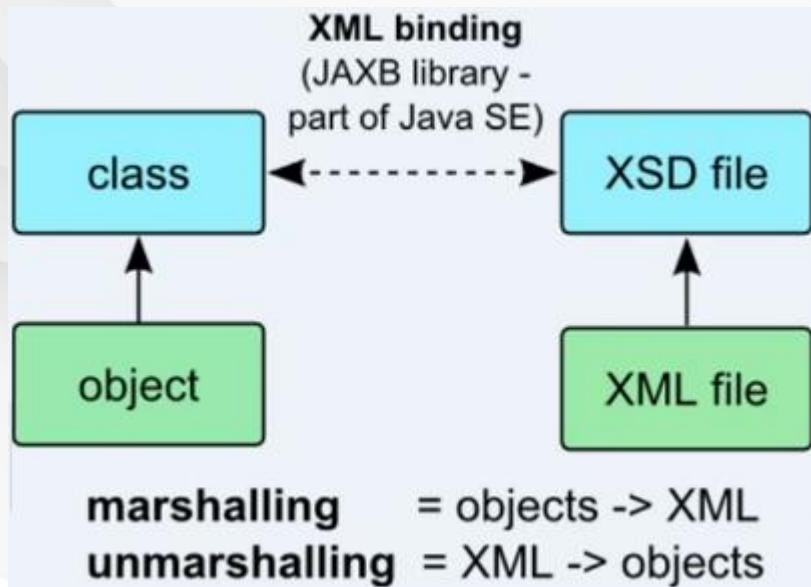
- XML 응답을 주는 RESTful 웹서비스 개요에 대하여 이해할 수 있습니다.
- XML 응답을 주는 RESTful 웹서비스 구현에 대하여 이해할 수 있습니다.

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner with a yellow decorative element on the left is positioned at the bottom of the image.

1. XML 응답을 주는 RESTful 웹서비스 개요

■ JAXB(Java Architecture for XML Binding)란?

JAXB는 Java 객체를 XML로 변환(직렬화, Marshalling) 해주고, XML을 Java 객체로 변환(역직렬화, Unmarshalling) 해주는 작업을 좀 더 편리하게 할 수 있도록 해주는 API이다.



■ 주요 JAXB 어노테이션(Annotation)

어노테이션	설명
@XmlRootElement	XML의 Root Element 이름을 정의한다. 클래스에서 사용하는 어노테이션으로 해당 클래스가 XML Root라는 것을 뜻합니다.
@XmlElement	XML의 Element 이름을 정의한다. 변수에 사용하는 어노테이션으로 해당 변수가 XML Element 라는 것을 뜻합니다.

■ JAXB를 사용한 RESTful 웹서비스 구현 절차

- ① Java 객체를 XML로 변환하기 위해 @XMLRootElement와 @XMLElement 어노테이션을 선언한 UserVOXML 클래스를 작성
- ② RestfulController 클래스에 사용자 목록을 조회하는 getUserListXml() 메서드를 작성하고 @RequestMapping과 @ResponseBody 어노테이션을 선언
- ③ Postman 툴을 이용해 메서드를 테스트
- ④ jQuery기반 Ajax통신을 하는 userList_Xml.html를 작성

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner with a yellow decorative element on the left is positioned at the bottom of the image.

2. XML 응답을 주는 RESTful 웹서비스 구현

Java객체를 XML로 변환 해주는 클래스 작성

UserVOXML.java

```
@XmlRootElement(name = "users")
public class UserVOXML {
    private String status;
    private List<UserVO> userList;

    public UserVOXML() {}

    public UserVOXML(String status, List<UserVO> userList) {
        this.status = status;
        this.userList = userList;
    }

    @XmlElement
    public void setStatus(String status) { this.status = status; }

    @XmlElement(name="user")
    public void setUserList(List<UserVO> userList) { this.userList = userList; }

    public String getStatus() { return status; }
    public List<UserVO> getUserList() { return userList; }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <status>success</status>
  <user>
    <city>제주</city>
    <gender>남</gender>
    <name>둘리</name>
    <userId>dooly</userId>
  </user>
  <user>
    <city>부산</city>
    <gender>남</gender>
    <name>박정우</name>
    <userId>jay</userId>
  </user>
  <user>
    <city>경기</city>
    <gender>여</gender>
    <name>박소울</name>
    <userId>vega2k</userId>
  </user>
</users>
```

■ RESTful 웹서비스 컨트롤러 클래스

```
+RestfulUserController.java ✕  
  
@Controller  
public class RestfulUserController {  
    @Autowired  
    private UserService userService;  
  
    @RequestMapping(value="/usersXml",  
        method=RequestMethod.GET)  
    @ResponseBody  
    public UserVOXML getUserListXml() {  
        List<UserVO> list = userService.getUserList();  
        UserVOXML xml = new UserVOXML("success", list);  
        return xml;  
    }  
}
```

Postman을 이용한 컨트롤러 클래스 테스트

GET http://localhost:8080/SpringWebPrj/usersXml

Params Send Save

Body Cookies Headers (4) Tests Status: 200 OK Time: 75 ms

Pretty Raw Preview XML

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <users>
3   <status>success</status>
4   <user>
5     <city>제주</city>
6     <gender>남</gender>
7     <name>돌리</name>
8     <userId>dooly</userId>
9   </user>
10  <user>
11    <city>부산</city>
12    <gender>남</gender>
13    <name>박정우</name>
14    <userId>jay</userId>
15  </user>
16  <user>
17    <city>경기</city>
18    <gender>여</gender>
19    <name>박소율</name>
20    <userId>vega2k</userId>
21  </user>
22 </users>
```

■ RESTful 웹서비스 클라이언트 html : Ajax 요청

```
userList_Xml.html ✕  
<script src="https://code.jquery.com/jquery-2.2.4.js"> </script>  
<script type="text/javascript">  
    $(function(){  
        $.ajax({  
            type:'get',  
            url:'usersXml',  
            contentType:'application/xml;charset=utf-8',  
            dataType:'xml',  
            error:function(xhr,status,msg){  
                alert("상태값 : " + status + " Http에러메시지 : "+msg);  
            },  
            success:showResult  
        });  
    });  
};
```

RESTful 웹서비스 클라이언트 html : Ajax 응답

userList_Xml.html

```
function showResult(xhr) {  
    console.log(xhr);  
    if($(xhr).find("status").text() == 'success') {  
        $(xhr).find("user").each(function(idx,user){  
            $('<tr>'  
                .append($('<td>').html($(user).find("userId").text()))  
                .append($('<td>').html($(user).find("name").text()))  
                .append($('<td>').html($(user).find("gender").text()))  
                .append($('<td>').html($(user).find("city").text()))  
                .appendTo('tbody');  
        });  
    }  
}  
//if  
//showResult  
</script>
```

```
<?xml version="1.0" encoding="UTF-8" ?>  
<users>  
    <status>success</status>  
    <user>  
        <city>제주</city>  
        <gender>남</gender>  
        <name>둘리</name>  
        <userId>doolly</userId>  
    </user>  
    <user>  
        <city>부산</city>  
        <gender>남</gender>  
        <name>박정우</name>  
        <userId>jay</userId>  
    </user>  
    <user>  
        <city>경기</city>  
        <gender>여</gender>  
        <name>박소울</name>  
        <userId>vega2k</userId>  
    </user>  
</users>
```

지금까지 [Spring RESTful 웹서비스 어플리케이션 작성(4)]에 대해서 살펴보았습니다.

XML 응답을 주는 RESTful 웹서비스 개요

- ◉ JAXB 개요
- ◉ @XmlRootElement, @XmlElement

XML 응답을 주는 RESTful 웹서비스 구현

- ◉ 서버 컨트롤러 클래스 작성
- ◉ 클라이언트 HTML 작성