스프링프레임웍 - Ajax 통신(@RequestBody, @ResponseBody) (/112)

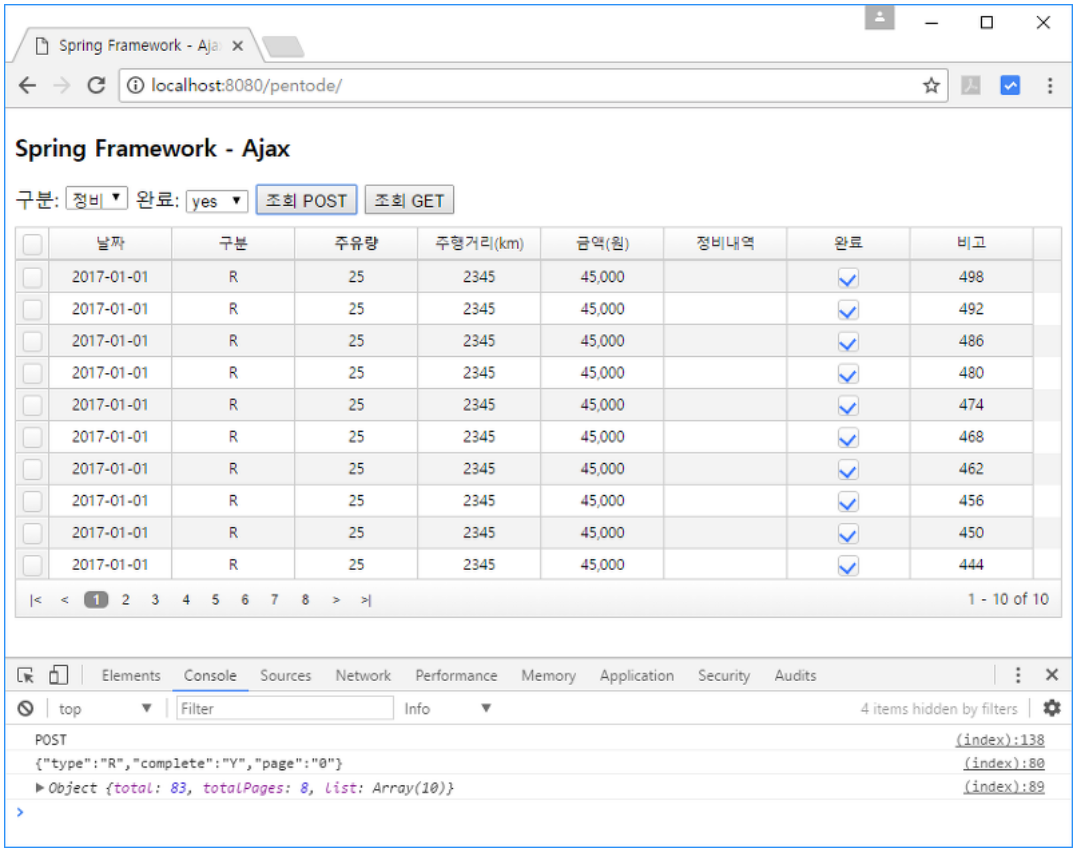
프로그래밍/스프링프레임워크 (/category/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D/%EC%8A%A4%ED%94%84%EB%A7%81%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81)

2018. 4. 9. 20:51

이번에는 스프링프레임웍을 사용해서 Ajax 통신을 하는 방법에 대해서 알아보도록 하겠습니다. jquery를 이용해서 Ajax로 데이터를 보내고, 결과를 JSON 데이터를 받는 예제입니다.

이번 예제는 **"스프링 프레임웍에서 MyBatis, Oracle 사용하기 (https://offbyone.tistory.com/18)"** 에서 사용된 프로젝트를 기반으로 합니다. 또한 클라이언트 측은 **"AX5UI - GRID 6 (페이징) (https://offbyone.tistory.com/111)"** 에서 사용된 예제를 변형하여 AX5UI 그리드에 Ajax와 JSON 데이터를 이용해서 조회하는 예제를 만들어 보겠습니다. 전체 소스를 하단에 첨부 되어 있습니다.

먼저 실행 결과 화면을 보겠습니다. AX5UI 그리드에 구분, 완료 값을 조회조건으로 하여 조회합니다. 조회는 POST 방식과 GET 방식일 경우 각각 어떻게 데이터를 전송하고 서버측에서 어떻게 처리하는지 알아볼 것입니다.



1. 테스트용 테이블과 데이터를 생성합니다.

차계부 테이블을 생성하고, 샘플 데이터 500개를 입력합니다. 나중에 조회에서 TYPE 과 COMPLETE 값을 조회조건으로하여 데이터를 조회해 볼 것입니다. 아래 sql 내용은 소스에 **doc/db.sql** 파일에도 있습니다.

```
CREATE TABLE TB_CHAGYEBU (
    NUM      INT NOT NULL,
    "DATE"   CHAR(10),
    TYPE     CHAR(1),
    AMOUNT   INT,
    MILEAGE  INT,
    PRICE    INT,
    REPAIR   VARCHAR(512),
    COMPLETE CHAR(1),
    NOTE     VARCHAR2(512)
);

ALTER TABLE TB_CHAGYEBU ADD CONSTRAINT PK_CHAGYEBU PRIMARY KEY(NUM);

DECLARE
   P_NUM INT := 1;
   P_TYPE CHAR(1);
   P_COMPLETE CHAR(1);
BEGIN
    LOOP
        IF MOD(P_NUM, 2) = 0 THEN
            P_TYPE := 'R';
        ELSE
            P_TYPE := 'O';
        END IF;
        IF MOD(P_NUM, 3) = 0 THEN
            P_COMPLETE := 'Y';
        ELSE
            P_COMPLETE := 'N';
        END IF;

        INSERT INTO TB_CHAGYEBU (NUM,"DATE",TYPE,AMOUNT,MILEAGE,PRICE,REPAIR,COMPLETE,NOTE) VALUES (P_NUM,'2017-01-01

        EXIT WHEN p_num >= 500;
        p_num := p_num + 1;
    END LOOP;
END;
/

COMMIT;
```

2. pom.xml 파일에 jackson 라이브러리를 추가 합니다.

이 라이브러리들은 JSON 데이터를 객체로 또는 그 반대로 맵핑하는데 사용됩니다.

```
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>2.8.8</version>
</dependency>

<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-annotations</artifactId>
    <version>2.8.8</version>
</dependency>

<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.8.8</version>
</dependency>
```

3. Rquest Mapping Handler와 View Resolver 설정

전에는 클라이언트로부터 보내온 JSON 형식의 데이터를 객체와 맵핑하기 위해서 **RequestMappingHandlerAdapter**에서 **MappingJackson2HttpMessageConverter** 메세지 컨버터를 사용하도록 설정을 했었습니다. 서버에서 조회된 데이터를 클라이언트의 Ajax 호출에 대한 응답으로 JSON 형태로 보내기 위해서는 **MappingJackson2JsonView** 를 **View Resolver**로 설정을 했었습니다. 그런데, **@RequestBody, @ResponseBody** 아노테이션을 사용하니 이런 설정 없이 처리 되었습니다. 결과적으로 요청 맵핑과 뷰 리졸버는 따로 설정하지 않아도 됩니다.(이 예제는 Spring 4 버전을 사용합니다.)

4. 클라이언트 측에서 GET 방식으로 조회 요청을 보내는 방법 입니다.(home.jsp)

AX5UI Grid의 자세한 설정방법은 "**AX5UI - GRID 6 (페이징) (https://offbyone.tistory.com/111)**" 를 참조 하세요.

```
function searchGet(_pageNo) {
    $('#page').val(_pageNo||0);

    //var sendData = {"type":$('#type').val(), "complete":$('#complete').val(), page:$('#page').val()};
    var sendData = $('#SearchForm').serialize();
    console.log(sendData);

    $.ajax({
        type: "GET",
        url : "<c:url value='/searchGet.do' />",
        data: sendData,
        async: true,
        success : function(data, status, xhr) {
            console.log(data);
            firstGrid.setData({
                list: data.list,
                page: {
                    currentPage: _pageNo,
                    pageSize: 10,
                    totalElements: data.total,
                    totalPages: data.totalPages
                }
            });
        },
        error: function(jqXHR, textStatus, errorThrown) {
            alert(jqXHR.responseText);
        }
    });
}
```

전송할 데이터를 두 가지 방법으로 만들 수 있습니다.

첫 번째는 직접 서버로 전송할 값들을 이용해서 데이터를 생성하는 방법입니다.

```
var sendData = {
    "type" : $('#type').val() ,
    "complete" : $('#complete').val() ,
    "page" : $('#page').val()
};
```

두 번째는 jquery 의 serialize() 메소드를 사용하는 방법입니다.

```
var sendData = $('#SearchForm').serialize();
```

두 가지중 어떤 방법을 사용하더라도 데이터는 자동으로 객체로 맵핑되어 집니다.

데이터 처리가 성공하면 success 에 붙여진 함수에 결과 데이터가 넘어옵니다.

data.list 는 그리드에 뿌려진 조회된 데이터 배열 입니다.
data.total 는 전체 데이터 수 입니다.

data.totalPages 는 전체 페이지 수 입니다.

5. 서버 측에서 GET 조회를 처리하는 방법 입니다.

HomeController.java

```
/**
 * GET 방식으로 값을 전달하는 방법 입니다.
 * @param vo
 * @return
 * @throws Exception
 */
@RequestMapping(value = "/searchGet.do", method = RequestMethod.GET)
public @ResponseBody Map<String, Object> searchGet(ChagyebuVO vo) throws Exception {
        LOGGER.info(vo.toString());

        return chagyebuService.selectChagyebuList(vo);
}
```

클라이언트로부터 보내진 데이터는 **searchGet(ChagyebuVO vo)** 에서 **ChagyebuVO** 객체에 자동으로 맵핑 됩니다. 아래의 서비스객체에서는 데이터 베이스에 조회를 하고, 결과를 Map 객체에 담아서 반환합니다. Map에는 조회 결과 "list" 와 전체 갯수 "total", 전체 페이지수 totalPages" 가 각각을 키로 데이터가 저장됩니다.

반환은 @ResponseBody Map<String, Object>에서 반환되는 전체 body 가 JSON 데이터로 변환 되어 전송 됩니다.

ChagyebuServiceImpl.java

```
@Override
@Transactional
public Map<String, Object> selectChagyebuList(ChagyebuVO vo) throws Exception {
        Map<String, Object> map = new HashMap<String, Object>();

        final int dataPerPage = 10;
        int page = vo.getPage();

        int first = page * dataPerPage + 1;
        int last = first + dataPerPage - 1;

        vo.setFirst(first);
        vo.setLast(last);

        LOGGER.info(vo.toString());

        Integer total = chagyebuDAO.selectChagyebuTotal(vo);
        Integer totalPages = (int)Math.ceil(total / dataPerPage);

        map.put("total", total);
        map.put("totalPages", totalPages);
        map.put("list", chagyebuDAO.selectChagyebuList(vo));

        return map;
}
```

6. 클라이언트 측에서 POST 방식으로 조회 요청을 보내는 방법 입니다.(home.jsp)

```
function searchPost(_pageNo) {
        $('#page').val(_pageNo||0);
        var sendData = JSON.stringify({type:$('#type').val(), complete:$('#complete').val(), page:$('#page').val()});
        console.log(sendData);
        $.ajax({
                type: "POST",
                url : "<c:url value='/searchPost.do' />",
                data: sendData,
                dataType: "json",
                contentType:"application/json;charset=UTF-8",
                async: true,
                success : function(data, status, xhr) {
                        console.log(data);
                    firstGrid.setData({
                    list: data.list,
                    page: {
                        currentPage: _pageNo,
                        pageSize: 10,
                        totalElements: data.total,
                        totalPages: data.totalPages
                    }
                });
                },
                error: function(jqXHR, textStatus, errorThrown) {
                        alert(jqXHR.responseText);
                }
        });
}
```

전송할 데이터를 직접 객체로 만들어서 JSON.stringify() 메소드로 문자열 화 합니다.

```
var sendData = JSON.stringify({type:$('#type').val(), complete:$('#complete').val(), page:$('#page').val()});
```

$.ajax 호출에서 **dataType**은 **"json"** 으로 **contentType**은 **"application/json;charset=UTF-8"** 으로 지정합니다. 데이터를 받는 부분은 GET 과 동일 합니다.

7. 서버 측에서 POST 방식의 조회를 처리하는 방법 입니다.

```
/**
 * POST 방식으로 값을 전달하는 방법 입니다.
 * @param vo
 * @return
 * @throws Exception
 */
@RequestMapping(value = "/searchPost.do", method = RequestMethod.POST)
public @ResponseBody Map<String, Object> searchPost(@RequestBody ChagyebuVO vo) throws Exception {
        LOGGER.info(vo.toString());

        return chagyebuService.selectChagyebuList(vo);
}
```

클라이언트에서 전송되는 데이터는 **searchPost(@RequestBody ChagyebuVO vo)** 에서 **@RequestBody** 를 사용해서 객체에 맵핑합니다. 조회된 결과는 GET 에서와 동일하게 **@ResponseBody Map<String, Object>** 처럼 반환값 지정에서 **@ResponseBody** 를 사용합니다.

이것으로 스프링프레임웍에서 Ajax 통신을 하는 방법을 알아보았습니다. 이번 예제에서는 GET 방식과 POST 방식을 모두 알아보았습니다. 이것을 응용하면 조회 뿐만 아니라 입력, 수정, 삭제 모두 해결할 수 있으리라 생각합니다.

※전체소스

spring_ajax.zip (https://offbyone.tistory.com/attachment/cfile1.uf@99CB8D455ACB53730F99CD.zip)

쉬고 싶은 개발자 구독하기

도움이 되셨다면 ♡공감 한번 꾹 ~ ^^ 로그인 하지 않아도 되요.

28 구독하기

Tag $.ajax (/tag/%24.ajax), @RequestBody (/tag/%40RequestBody), @ResponseBody (/tag/%40ResponseBody), ajax (/tag/ajax), get (/tag/get), POST (/tag/POST), springframework (/tag/springframework),
스프링프레임웍 (/tag/%EC%8A%A4%ED%94%84%EB%A7%81%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8D)

댓글 2개가 달렸습니다

댓글을 달아 주세요

노그인 *2019.09.25 12:40* 댓글주소 (/112#comment5567082) 수정/삭제 댓글쓰기
로그인하지 않아도 되서 공감 박고 갑니다ㅋㅋ

pentode (https://offbyone.tistory.com) *2019.09.25 14:01 신고 (/toolbar/popup/abuseReport/?entryId=112&commentId=5567115)* 댓글주소 (/112#comment5567115)
수정/삭제
방문해 주셔서 감사합니다.^^

이름

비밀번호

http://

비밀글

댓글 달기

« (/113) 1 (/412) ··· 294 (/116) 295 (/115) 296 (/114) 297 (/113) 298 299 (/111) 300 (/110) 301 (/109) 302 (/108) ···
408 (/1) » (/111)

검색어를 입력하세요 검색

공지사항

- 블로그 이전을 완료.. (/notice/4)

카테고리

- 분류 전체보기 (408) (/category)
  - 프로그래밍 (389) (/category/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D)
    - PHP (19) (/category/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D/PHP)

태그목록

코틀린 (/tag/%EC%BD%94%ED%8B%80%EB%A6%B0), 자바 (/tag/%EC%9E%90%EB%B0%94), Kotlin (/tag/Kotlin), php (/tag/php), mysql (/tag/mysql), 전자정부표준프레임워크 (/tag/%EC%A0%84%EC%9E%90%EC%A0%95%EB%B6%80%ED%91%9C%EC%A4%80%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC), 스프링프레임워크 (/tag/%EC%8A%A4%ED%94%84%EB%A7%81%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC), 이클립스 (/tag/%EC%9D%B4%ED%81%B4%EB%A6%BD%EC%8A%A4), eclipse (/tag/eclipse), Windows 10 (/tag/Windows%2010), MariaDB (/tag/MariaDB), 안드로이드 (/tag/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C), 오라클 (/tag/%EC%98%A4%EB%9D%BC%ED%81%B4), AX5UI (/tag/AX5UI), springframework (/tag/springframework), 스프링 프레임워크 (/tag/%EC%8A%A4%ED%94%84%EB%A7%81%20%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC), jQuery (/tag/jQuery), 엑셀 (/tag/%EC%97%91%EC%85%80), 안드로이드 스튜디오 (/tag/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C%20%EC%8A%A4%ED%8A%9C%EB%94%94%EC%98%A4), 전자정부 표준프레임워크 (/tag/%EC%A0%84%EC%9E%90%EC%A0%95%EB%B6%80%20%ED%91%9C%EC%A4%80%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC), 파이썬 (/tag/%ED%8C%8C%EC%9D%B4%EC%8D%AC), java (/tag/java), Spring Framework (/tag/Spring%20Framework), PYTHON (/tag/PYTHON), iPhone 5s (/tag/iPhone%205s), Oracle (/tag/Oracle), jquery UI (/tag/jquery%20UI), spring boot (/tag/spring%20boot), Android Studio (/tag/Android%20Studio), centos 7 (/tag/centos%207)

링크

- LOVE LETTER (http://deborah.tistory.com/)
- PENTODE의 日常 (http://pentode.tistory.com)

pentode

웹 프로그래밍에 사용되는 유용한 팁을 공유합니다.

LATEST FROM OUR BLOG

LATEST COMMENTS

- [승인대기]. (/230#comment5630892) **걱정하지않는자** *12.25*
- 도움이 되셨다니 저... (/336#comment5627780) **pentode** *12.22*
- 업드려 절!! 감사합... (/336#comment5627354) **맥심매니아** *12.21*
- 자동으로 생성되는... (/291#comment5625735) **pentode** *12.17*
- URL 뒤에 붙는 Slas... (/334#comment5625733) **pentode** *12.17*
- 내용 중 궁금한게... (/291#comment5625587) **궁금** *12.17*
- 개발자님 문의좀 드... (/334#comment5625541) **개심플** *12.17*
- 감사합니다. 수정했... (/364#comment5624050) **pentode** *12.13*
- 2017년 업데이트 이... (/104#comment5624048) **pentode** *12.13*
- Ctrl이 아니라 Alt... (/364#comment5623502) **ㄴㅇㄹ** *12.12*