(디지털컨버전스)스마트 콘텐츠와 웹 응용응용SW개발자 양성과정

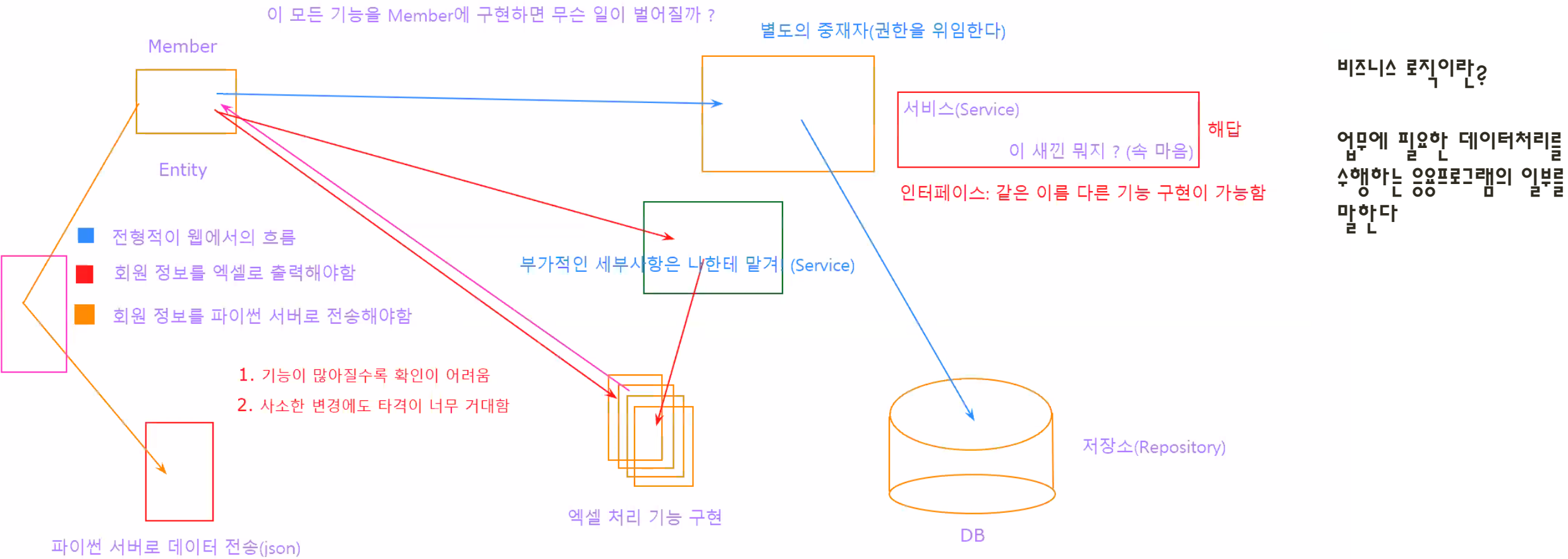
강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 –Joongyeon Kim(김중연)
jjjr69@naver.com

2021년 6월 21일 질문노트

[김중연]

```
{
  "pageno": 0,
  "pagesize": 0,
  "totalcount": 100,
  "contacts": [
    {
      "no": 1592030163205,
      "name": "Keandra Morris",
      "tel": "010-3456-8299",
      "address": "서울시",
      "photo": "http://sample.bmaster.kro.kr/photos/100.jpg"
    },
    {
      "no": 1592030163204,
      "name": "Katherine Long",
      "tel": "010-3456-8298",
      "address": "서울시",
      "photo": "http://sample.bmaster.kro.kr/photos/99.jpg"
    },
    {
      "no": 1592030163203,
      "name": "Katchi Hughes",
      "tel": "010-3456-8297",
      "address": "서울시",
      "photo": "http://sample.bmaster.kro.kr/photos/98.jpg"
    },
    {
      "no": 1592030163202,
      "name": "Kalisa Lee",
      "tel": "010-3456-8296",
      "address": "서울시",
      "photo": "http://sample.bmaster.kro.kr/photos/97.jpg"
    },
```


JSON의 형태 일단 이렇게 생겼다는 것만 알아두고 가자



비즈니스 로직이란?

업무에 필요한 데이터처리를 수행하는 응용프로그램의 일부를 말한다

엑셀 처리 기능이라는 하위의 세부사항에 Member 라는 녀석이 종속된 상태이기 때문

DB를 생성할 때
Entity는 DB에 직접 연결된다(즉 함부로 값을 변경해서는 안된다!)
Repository는 DB에 접근하는 모든 코드가 모여있다
Service는 DB에 접근하는 코드를 repository에 위임하고 비즈니스 로직과 관련된 모든 코드가 모여있다
이렇게 구분해두면 비즈니스 로직과 관련된 부분에 문제가 발생했을 때는 service 패키지를 확인하고, DB 접근과 관련된 문제가 발생하면 Repository 부분을 확인하면 된다

```java
package com.example.demo.controller.board;

import lombok.extern.slf4j.Slf4j;
import org.json.JSONArray;
import org.json.JSONObject;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@Slf4j
// Controller vs RestController의 차이점은 딱 하나다.
// Controller는 HTML을 파싱하는 반면
// RestController는 json을 파싱한다.
@RestController
public class SixthController {

    @GetMapping("/jsonTest")
    public String getJsonTest () {

        return "I'm JSON!!!!";
    }

    @GetMapping("/jsonKeyValueTest")
    public String getJsonKeyValueTest () {
        // 모든 json 데이터는 아래와 같이 생겼음
        // 향후 Vue에서 보내는 데이터도 전부 이 json 형태로 날아옴
        // 우리가 SRT API나 여러가지 Rest API라고 하는 녀석들도
        // 요청을 하면 데이터 형식이 전부 json 형식으로 날아옴
        String jsonString = "{\"title\":\"hihi\"," +
                "\"draft\":false," +
                "\"star\":5" +
                "}";
```

```
    // JSON 파싱
    JSONObject jObj = new JSONObject(jsonString);
    String title = jObj.getString( key: "title");
    Boolean draft = jObj.getBoolean( key: "draft");
    Integer star = jObj.getInt( key: "star");

    log.info("title: " + title + ", draft: " + draft + ", star: " + star);

    return jsonString;
}

@GetMapping("/jsonMultiObjectTest")
public String getJsonMultiObjectTest () {  //이중 HashMap이랑 비슷한 구조다
    String jsonString = "{" +
            "\"movie1\": {" +
            "\"title\": \"hihi\", " +
            "\"draft\": false, " +
            "\"star\": 5" +
            "}, " +
            "\"movie2\": {" +
            "\"title\": \"code monkey\", " +
            "\"draft\": false, " +
            "\"star\": 5" +
            "}" +
            "}";

    // json 내에 Object 형식이 구성된 경우의 파싱
    JSONObject jObj = new JSONObject(jsonString);

    //JSON코드의 값을 출력함
    JSONObject movie1Obj = jObj.getJSONObject("movie1");
```

```
    log.info("movie1: " + movie1Obj.toString() +
            "\ntitle: " + movie1Obj.getString( key: "title") +
            ", draft: " + movie1Obj.getBoolean( key: "draft") +
            ", star: " + movie1Obj.getInt( key: "star"));

    JSONObject movie2Obj = jObj.getJSONObject("movie2");
    log.info("movie2: " + movie2Obj.toString() +
            "\ntitle: " + movie2Obj.getString( key: "title") +
            ", draft: " + movie2Obj.getBoolean( key: "draft") +
            ", star: " + movie2Obj.getInt( key: "star"));

    return jsonString;
}

// 현재 시점까지의 json 처리는
// 향후 Vue에서 AXIOS라는 것을 사용하며 모두 처리가 될 것임
// 한 가지 차이가 있다면 나중에 python과 연동할 때
// Spring의 Requester를 통해서
// Spring 자체가 Client가 되어 Python Server에 요청을 넣어야함
// 그 시점에서는 Spring이 직접 JSON을 처리할 필요가 있음
@GetMapping("/jsonArrayTest")
public String getJsonPowerTest () {
    String jsonString = "{" +
            "\"movies\": [" +
            "{" +
            "\"title\": \"hihi\", " +
            "\"draft\": false, " +
            "\"star\": 5" +
            "}, " +
            "{" +
            "\"title\": \"code monkey\", " +
```

```
            "\"draft\": false, " +
            "\"star\": 5" +
            "}, " +
            "{" +
            "\"title\": \"monkey magic\", " +
            "\"draft\": false, " +
            "\"star\": 4.7" +
            "}" +
            "]" +
            "}";

    // JSON 배열 파싱
    JSONObject jObj = new JSONObject(jsonString);
    JSONArray jArr = jObj.getJSONArray( key: "movies");

    // 루프를 돌며 JSON 배열의 모든 정보를 출력함
    for (int i = 0; i < jArr.length(); i++) {
        JSONObject obj = jArr.getJSONObject(i);

        //위에 있는 JSON코드의 값을 설정해줌
        String title = obj.getString( key: "title");
        Boolean draft = obj.getBoolean( key: "draft");
        Float star = obj.getFloat( key: "star");

        log.info("title: " + title + ", draft: " + draft + ", star: " + star);
    }

    return jsonString;
}
}
```