


SeSAC 4기, 

DAO, DTO, VO 수업

WITH 팀 리뷰드



DAO

# DAO 란?

- Data Access Object
- DB의 data에 접근하기 위한 객체로 직접 DB에 접근하여 데이터를 삽입, 삭제, 조회 등 조작할 수 있는 기능을 수행한다.
- Spring 에서 Repository package 가 바로 DAO

DTO

# DTO 란?

- Data Transfer Object
- 계층 간 데이터 교환을 위해 사용하는 객체
- 다른 로직을 가지지 않는 순수한 데이터 객체 ( Java Beans )

```
import lombok.Getter;  
import lombok.Setter;  
  
@Getter  
@Setter  
public class DTOExample{  
    private String title;  
    private String content;  
    private String writer;  
}
```

# DTO란?

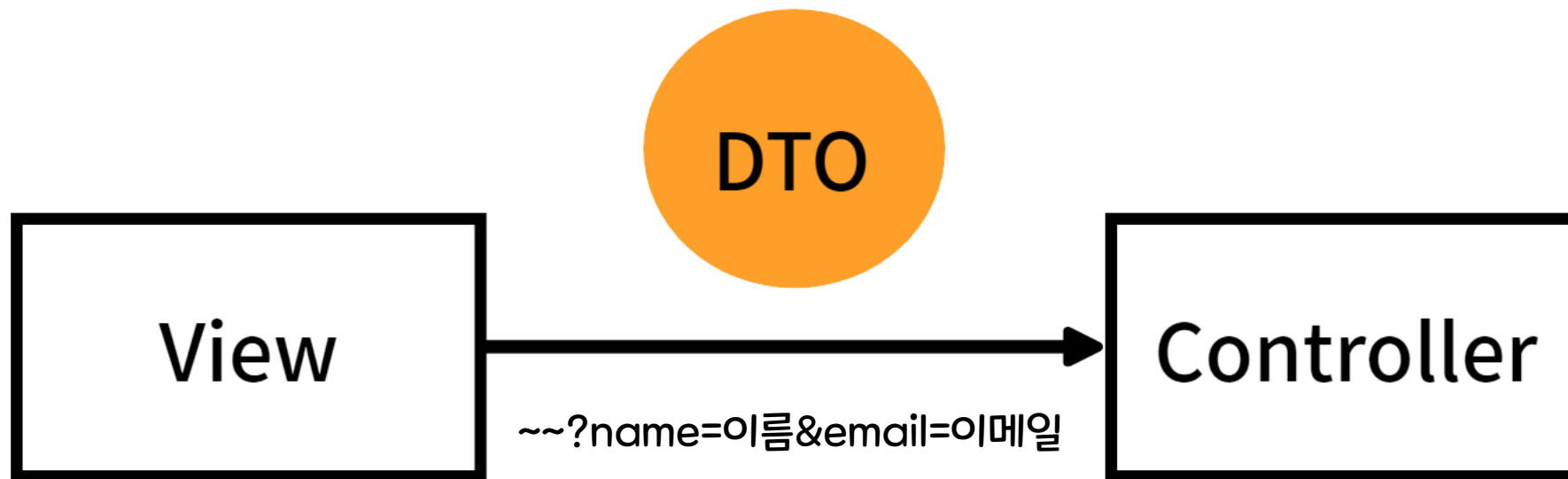
```
import lombok.Getter;  
import lombok.Setter;
```

```
@Getter  
@Setter
```

```
public class DTOExample{  
    private String title;  
    private String content;  
    private String writer;  
}
```

```
public String getTitle() {  
    return title;  
}  
  
public void setTitle(String title) {  
    this.title = title;  
}  
  
public String getContent() {  
    return content;  
}  
  
public void setContent(String content) {  
    this.content = content;  
}  
  
public String getWriter() {  
    return writer;  
}  
  
public void setWriter(String writer) {  
    this.writer = writer;  
}
```

# GET 전송 시 DTO 사용하기

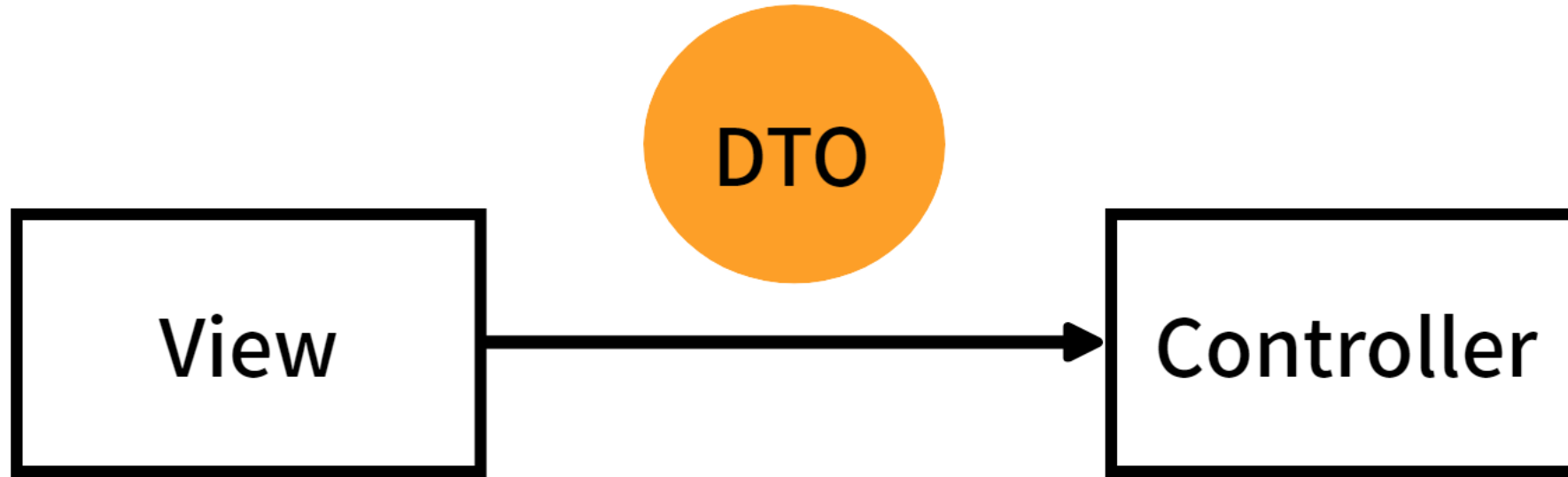


```
@GetMapping(value = "/request3")
public String getRequestParam3(MemberDTO memberDTO) {
    //return memberDTO.getName() + " " + memberDTO.getEmail() + " "
    return memberDTO.toString();
}

public class MemberDTO {
    private String name;
    private String email;
    ...
}
```



# POST 전송 시 DTO 사용하기



```
@PostMapping("mvc-post")
public String postMVC(
    @RequestParam(value = "name", required = false) String name,
    @RequestParam(value = "age", required = false) String age,
    Model model) {
    model.addAttribute(attributeName: "name", name);
    model.addAttribute(attributeName: "age", age);
    return "04_API";
}
```

```
@PostMapping("mvc-post")
public String postMVC(
    PostDTO postDTO,
    Model model) {
    model.addAttribute(attributeName: "name", postDTO.getName());
    model.addAttribute(attributeName: "age", postDTO.getAge());
    return "04_API";
}
```

# 실습 5. DTO 사용 실습

폼 전송 - 실습

이름

성별 ☐ 남자 ☐ 여자

생년월일  년  월  일

관심사 ☐ 여행 ☐ 패션 ☐ 음식

회원가입

이름 : 홍길동

성별 : 남자

생년월일 : 1976-5-15

관심사 : 여행, 패션, 음식

1. 실습 4에서 진행했던 form 전송을 view에서 보낸 데이터를 DTO로 받아 처리하기

VO

# VO란?

- Value Object
- DTO와 비슷하지만, VO는 read-Only 속성을 갖고 있는 객체
- Getter 만 가지고 있어 값에 대한 수정이 불가능하다.

**동적폼전송**

# 실습 6. 동적폼전송 실습

Axios를 이용해 동적폼전송 구현하기

전송

폼 전송 - 실습

이름 김규리

성별 ☐ 남자 ☒ 여자

생년월일 1950 ▼ 년 1 ▼ 월 1 ▼ 일

관심사 ☒ 여행 ☐ 패션 ☒ 음식

DevTools is now available in Korean! Always match

Elements Console Sources Ne

top ▼ Filter

김규리회원가입 성공

> |

# @RequestBody

- View 에서 Controller로 값을 전달할 때 HTTP Body 안에 JSON 을 VO에 매핑시켜주는 Spring Annotation

```
import lombok.Getter;

2 usages
@Getter
public class IntroduceVO {
    private String name;
    private String gender;
    private int year;
    private int month;
    private int day;
    private String interest;
}
```

```
public String postIntroduce6(
    @RequestBody IntroduceVO introduceVO
) {
```

- IntroduceVO 에는 Setter이 없음에도 값 이 할당된다.



# 실습 7. 동적폼전송 @RequestBody

전송

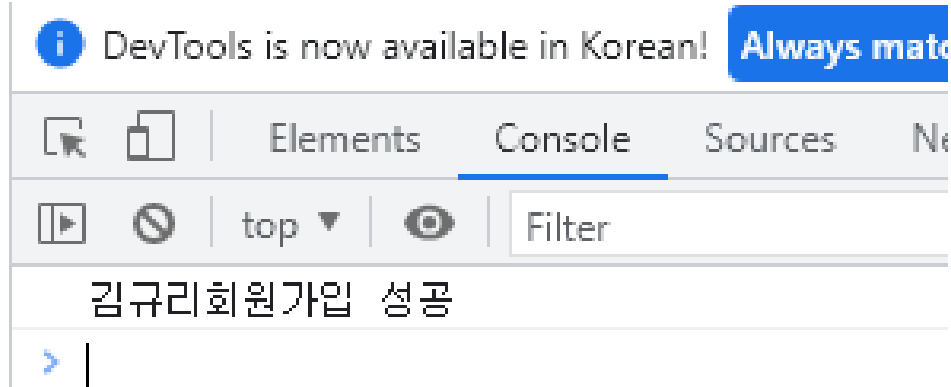
폼 전송 - 실습

이름 김규리

성별 ☐ 남자 ☒ 여자

생년월일 1950 ▼ 년 1 ▼ 월 1 ▼ 일

관심사 ☒ 여행 ☐ 패션 ☒ 음식



- 실습 6에서 진행했던 실습을 DTO 가 아닌 VO에서 동작 가능하게 만들기

# 실습 8. 회원관리 시스템

- C : 회원 가입
  - R : 로그인
  - U : 회원 정보 수정
  - D : 회원 탈퇴
- 
- 위의 기능을 수행할 수 있는 RESTful 한 시스템을 만들어보시오.