

VO vs DTO

2017. 1. 12. 16:21

[DTO](#) [java](#) [Vo](#)

VO vs DTO

VO(Value Object)

- 데이터 그 자체로 의미 있는 것을 담고 있는 객체이다.
- DTO와 동일한 개념이나 차이점은 Read-Only 속성 객체이다.
- 간단한 독립체(Entity)를 의미하는 작은 객체를 의미한다.
- 관계데이터베이스의 레코드에 대응되는 자바클래스이다.
 - 형태는 Database레코드를 구성하는 필드들을 VO의 Attribute로 하고 해당 변수에 접근 할 수 있는 Getter와 Setter 메소드의 조합으로 클래스를 형성되어진 클래스이다.
 - 거의 불변성을 가지고 equals()로 비교할 때 객체의 모든 값을 비교해야 한다.

색상으로 예를 들어보겠다. 색상 중에 빨강과 초록을 RGBA로 나타내면RGBA(255,0,0,0)와 RGBA(0,255,0,0) 로 표현된다. 더 나아가서 코드명이 붙은색상도 존재할 거이다. 코드로 보면 아래와 같다.

```

1 class Color {
2
3 private int R,G,B,A;
4
5 public Color(int r,int g, int b, int a){...}
6
7 //getters and setters
8
9 public final static Color RED = new Color(255,0,0);
10
11 public final static Color GREEN = new Color(0,255,0,0);
12
13 }

```

Colored by Color Scripter

위와 같은 색상 객체가 있고 빨강과 초록이 있다. 코드에서 빨강 또는 초록을 사용할 때, Color.RED, Color.GREEN 를 사용하면 된다. 즉, 값 객체 Color.RED, Color.GREEN 자체로서 의미가 있게 된다.

DTO(Data Transfer Object)

- 전송되는 데이터의 컨테이너이다.
- VO와 동일하게 데이터를 저장하여 사용하도록 하는 부분에서 필요하다.
- VO와 비교를 하여 보면 DTO는 같은 시스템에서 사용되는 것이 아닌 다른 시스템으로 전달하는 작업을 처리하는 객체이다.
- Layer간의 통신 용도로 오가는 객체를 말하기도 한다.

- 현재의 개발 환경에서 보통 데이터는 다음과 같이 흐름으로 이동한다.
 - 서버 측 : Database Column Data -> DTO -> API(JSON or XML) -> Client
 - 클라이언트 측 : Server -> API(JSON or XML) -> DTO -> View or Local Database System

VO와 DTO 정리 :

- VO : 사용 되는 값이 객체로 표현 되며, 값 변경이 없는 경우를 말한다.
- DTO : 데이터의 전송을 위한 객체이며, 비즈니스 로직까지 담아서 사용하기 한다.

그런데 VO/DTO에 대한 논쟁도 많고 반드시 사용 용도를 구분하는 건 무의미해 보인다. 개발팀 내에서 용도를 정해서 사용하면 좋을 것 같다.

예를 들어 외부 시스템과 데이터 통신을 할 경우에는 DTO로, DB에서 가져오는 Data는 VO로 정의해서 사용한다고 약속을 하면, 향후 DTO/VO를 수정할 경우 좀 더 고민할 여지가 생길것 같다.

참고 사이트

<http://mrgamza.tistory.com/49>

<https://martinfowler.com/bliki/ValueObject.html>

<http://parkgaram.com/blog/vo-%EC%99%80-dto-%EC%B0%A8%EC%9D%B4/>