Chapter 6-4 Validation

```
public class PostDto {
    private int id;
    private String title;
    private String content;
    private String writer;
    private int boardId;
```

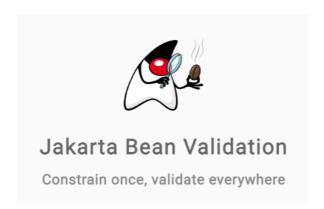
제목: 비어있지 않아야 함 본문: 400자 이하 작성자: 3자 이상 10자 이하

사용자의 입력은 어떻게 들어올지 예측하기 어렵다.

```
public void createPost(PostDto postDto){
    if (postDto.getTitle() == null) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST);
    }
    if (postDto.getTitle().isBlank()) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST);
    }
    if (postDto.getContent().length() > 400) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST);
    }
    this postDag createPost(postDto);
```

소스 코드 상에서 일일이 확인하는 것은 매우 고되다.

Chapter 6-4 Validation 1



Java Bean Validation - 객체 상의 제약사항의 명세를 제공하기 위한 프로젝트



JPA와 Hibernate ORM과의 관계와 비슷하다.

```
8. Built-in Constraint definitions
 8.1. @Null constraint
  8.2. @NotNull constraint
  8.3. @AssertTrue constraint
  8.4. @AssertFalse constraint
                                              // Null 여부
                                                                                         // String 또는 Collection의 크기
  8.5. @Min constraint
                                              @Null
 8.6. @Max constraint
                                                                                         @Size
 8.7. @DecimalMin constraint
                                              @NotNull
  8.8. @DecimalNax constraint
  8.9. @Negative constraint
  8.10. @NegativeOrZero constraint
  8.11. @Positive constraint
  8.12. @PositiveOrZero constraint
  8.13. @Size constraint
                                                          // String 또는 Collection의 Size > 0 인지
  8.14. @Digits constraint
                                                          @NotEmpty
  8.15. @Past constraint
  8.16. @PastOrPresent constraint
  8.17. @Future constraint
                                                          // String이 공백을 제외한 문자열인지
  8.18, @FutureOrPresent constraint
 8.19. @Pattern constraint
                                                          @NotBlank
  8.20. @NotEmpty constraint
  8.21. @NotBlank constraint
 8.22. @Email constraint
```

Chapter 6-4 Validation 2

19: 정규표현식에 대한 검사

```
@NotNull변수가 null이냐 아니냐@NotEmptynull이 아니면서 Object.size() > 0@NotBlank공백이 아닌 문자열 " "
```

```
private int id;
@NotNull(message = "title not null")
private String title;
@Size(max = 40, message = "size under 40")
private String content;
@Size(min = 3, max = 10, message = "size between 3 -10")
private String writer;
private int boardId;
```

Chapter 6-4 Validation 3