

## Bean Validation

Annotation	Constraint	Example
@Null	Null인지 검증	@Null
@NotNull	Null이 아님을 검증 공백 문자열은 OK	@NotNull String title;
@Max	정수 최대 값	@Max(100) int score;
@Min	정수 최소 값	@Min(18) int age;
@Size	문자열 혹은 Collection 프레임 워크가 지정한 범위의 크기인 지 검증	@Size(min=0, max=10) List<BoardVO> list;
@AssertTrue	True 인가?	@AssertTrue public Boolean insertBoard( ){ }
@AssertFalse	False 인가?	@AssertFalse public Boolean insertBoard( ){ }
@Pattern	정규표현식을 만족하는지 검증	@Pattern("[a-z][0-9]+") String id;
@DecimalMax	실수 최대 값	@DecimalMax(10.0) double score;
@DecimalMin	실수 최소 값	@DecimalMin(0.0) double area;

<!-- Bean Validation API -->

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.1.0.Final</version>

</dependency>

Bean Validation Specification : <http://jcp.org/en/jsr/detail?id=303>

## Hibernate Validator

Annotation	Constraint	Example
@NotEmpty	문자열이나 Collection 프레임워크가 null 또는 빈 문자열인지 검증. 문자열 전후의 공백을 제거하지 않음.	@NotEmpty List<BoardVO> list;
@NotBlank	문자열이 null 또는 비어있는지 검증. 문자열 전후의 공백 제거.	@NotBlank String id;
@Length	범위 내 길이의 문자열인지 검증.	@Length(min=0, max=12) String pwd;
@Range	범위 내의 숫자인지 검증	@Range(min=80, max=100) int score;
@Email	이메일 형식인지 검증	@Email String email;
@URL	URL 형식인지 검증	@URL String url;

```
<!-- Hibernate Validator Engine -->
```

```
<dependency>
```

```
    <groupId>org.hibernate</groupId>
```

```
    <artifactId>hibernate-validator</artifactId>
```

```
    <version>5.1.0.Alpha1</version>
```

```
</dependency>
```

Hibernate Validator : <http://hibernate.org/subprojects/validator>