Spring Boot 유효성 검사 (Jakarta Bean Validation)

> Spring Boot 4.x / Jakarta Validation 3.0 기준 (2026 최신)

1. 의존성 추가

`build.gradle`에 한 줄 추가:

```
implementation 'org.springframework.boot:spring-boot-starter-validation'
```

> spring-boot-starter-validation에 **Hibernate Validator** (Jakarta Validation 구현체)가 포함되어 있다.

---

2. DTO에 어노테이션 붙이기

주요 어노테이션

어노테이션	설명	예시
`@NotNull`	null 불가	객체 타입 필드
`@NotEmpty`	null, "" 불가	String, Collection
`@NotBlank`	null, "", " " 불가	String (공백도 차단)
`@Size(min, max)`	길이/크기 제한	`@Size(min=2, max=20)`
`@Min` / `@Max`	숫자 최소/최대	`@Min(1)`
`@Email`	이메일 형식	`@Email`
`@Pattern`	정규식 매칭	`@Pattern(regexp="...")`

적용 예시 - BoardRequest

```
package com.example.boardv1.board;

import jakarta.validation.constraints.NotBlank;
import lombok.Data;

public class BoardRequest {

    @Data
    public static class SaveOrUpdateDTO {
        @NotBlank(message = "제목을 입력해주세요")
        private String title;

        @NotBlank(message = "내용을 입력해주세요")
        private String content;
```

```
    }
}
```

적용 예시 - UserRequest

```
package com.example.boardv1.user;

import jakarta.validation.constraints.Email;
import jakarta.validation.constraints.NotBlank;
import jakarta.validation.constraints.Size;
import lombok.Data;

public class UserRequest {

    @Data
    public static class LoginDTO {
        @NotBlank(message = "유저네임을 입력해주세요")
        private String username;

        @NotBlank(message = "비밀번호를 입력해주세요")
        private String password;
    }

    @Data
    public static class JoinDTO {
        @NotBlank(message = "유저네임을 입력해주세요")
        @Size(min = 3, max = 20, message = "유저네임은 3~20자로 입력해주세요")
        private String username;

        @NotBlank(message = "비밀번호를 입력해주세요")
        @Size(min = 4, max = 20, message = "비밀번호는 4~20자로 입력해주세요")
        private String password;

        @Email(message = "이메일 형식이 올바르지 않습니다")
        private String email;
    }
}
```

---

3. Controller에서 @Valid 붙이기

DTO 파라미터 앞에 `@Valid`만 붙이면 자동으로 유효성 검사가 실행된다.

```
import jakarta.validation.Valid;
import org.springframework.validation.Errors;

@PostMapping("/boards/save")
public String save(@Valid BoardRequest.SaveOrUpdateDTO reqDTO, Errors errors) {
```

```
    // 유효성 검사 실패 시
    if (errors.hasErrors()) {
        throw new Exception400(errors.getAllErrors().get(0).getDefaultMessage());
    }

    User sessionUser = (User) session.getAttribute("sessionUser");
    if (sessionUser == null)
        throw new Exception401("인증되지 않았습니다.");

    boardService.게시글쓰기(reqDTO.getTitle(), reqDTO.getContent());
    return "redirect:/";
}
```

핵심 포인트

항목	설명
`@Valid`	DTO 파라미터 앞에 붙여서 유효성 검사 활성화
`Errors`	`@Valid` 바로 뒤에 위치해야 함 (순서 중요!)
`errors.hasErrors()`	유효성 검사 실패 여부 확인
`errors.getAllErrors().get(0).getDefaultMessage()`	첫 번째 에러 메시지 추출

> **순서 주의**: `@Valid DTO dto, Errors errors` - Errors는 반드시 @Valid 파라미터 바로 다음에 와야 한다.

---

4. GlobalExceptionHandler에서 처리

이 프로젝트는 이미 `Exception400`과 `GlobalExceptionHandler`가 있으므로 별도 추가 없이 동작한다.

```
유효성 검사 실패 → errors.hasErrors() → throw Exception400("메시지")
    → GlobalExceptionHandler.ex400() → alert + history.back()
```

기존 코드가 그대로 활용된다:

```
@ExceptionHandler(exception = Exception400.class)
public String ex400(Exception400 e) {
    String html = String.format("""
            <script>
                alert('%s');
                history.back();
            </script>
            """, e.getMessage());
    return html;
}
```

5. 전체 흐름 요약

```
[요청]
사용자가 폼 제출 (title="", content="내용")
    ↓
[Controller]
@Valid SaveOrUpdateDTO reqDTO, Errors errors
    ↓
[Jakarta Validation]
@NotBlank 검사 → title이 빈 문자열 → 실패
    ↓
[에러 확인]
errors.hasErrors() == true
    ↓
[예외 발생]
throw new Exception400("제목을 입력해주세요")
    ↓
[GlobalExceptionHandler]
ex400() → alert('제목을 입력해주세요') + history.back()
    ↓
[화면]
알림창 표시 → 이전 페이지로 돌아감
```

6. 적용 체크리스트

```
1. build.gradle에 spring-boot-starter-validation 추가
2. DTO 필드에 @NotBlank, @Size 등 어노테이션 추가
3. Controller 메서드에 @Valid + Errors 파라미터 추가
4. errors.hasErrors() 체크 후 Exception400 throw
```

변경 파일 목록

파일	변경 내용
`build.gradle`	`implementation 'org.springframework.boot:spring-boot-starter-validation'` 추가
`BoardRequest.java`	DTO 필드에 `@NotBlank` 추가
`UserRequest.java`	DTO 필드에 `@NotBlank`, `@Size`, `@Email` 추가
`BoardController.java`	`@Valid` + `Errors` 파라미터 추가, 에러 체크 로직 추가
`UserController.java`	`@Valid` + `Errors` 파라미터 추가, 에러 체크 로직 추가

참고: @Valid vs @Validated

항목	@Valid	@Validated
패키지	`jakarta.validation`	`org.springframework.validation.annotation`
표준	Jakarta 표준 (JSR 380)	Spring 전용
그룹 검증	미지원	지원
사용 위치	메서드 파라미터, 필드	클래스, 메서드 파라미터
권장	일반적인 DTO 검증	그룹별 검증이 필요할 때

> 이 프로젝트에서는 `@Valid`만으로 충분하다.