

Lombok is a Java library that can generate known patterns of code for us, allowing us to reduce code. In this article, we'll walk through each of the 16 annotations provided by Lombok.

To Use Lombok you must add dependancy

```
<dependency>  
    <groupId>org.projectlombok</groupId>  
    <artifactId>lombok</artifactId>  
</dependency>
```

Spring Boot provides validation, including annotations, custom validators, error handling and group validation.

To Use **validation** you must add dependancy

```
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-validation</artifactId>  
</dependency>
```

Spring Boot validation

@NotNull: Ensures the value is not null.

```
java

@NotNull
private String name;
```

@Size: Specifies the size of a string or collection.

```
java

@Size(min = 2, max = 30)
private String username;
```

@Min: Ensures the value is greater than or equal to a specified minimum.

```
java

@Min(18)
private Integer age;
```

@Max: Ensures the value is less than or equal to a specified maximum.

```
java

@Max(100)
private Integer score;
```

@Email: Validates that the string is a valid email address.

```
java

@email
private String email;
```

@Pattern: Validates the string against a specified regex pattern.

```
java

@Pattern(regexp = "[A-Za-z0-9]+$")
private String username;
```

@Past: Ensures that the date is in the past.

```
java

@Past
private LocalDate birthDate;
```

Spring Boot validation

@Future: Ensures that the date is in the future.

```
java

@Future
private LocalDate eventDate;
```

@NotEmpty: Ensures that the string is not null and not empty.

```
java

@NotEmpty
private String address;
```

@NotBlank: Ensures that the string is not null and not whitespace.

```
java

@NotBlank
private String password;
```

object mapping

object mapping (converting between different object types) is commonly done using libraries like **MapStruct** or **ModelMapper**.

To Use **mapper** you must add dependancy

```
<dependency>  
  <groupId>org.modelmapper</groupId>  
  <artifactId>modelmapper</artifactId>  
  <version>2.4.5</version>  
</dependency>
```

`@Mapper` is an annotation used in MapStruct, a Java library that simplifies the mapping between Java bean types. When you annotate an interface with `@Mapper`, MapStruct generates an implementation of that interface at compile time, providing a way to easily convert between different object types (like DTOs and entities).

1. **200 OK:** The request was successful, and the server returned the requested data.
2. **201 Created:** The request was successful, and a new resource was created (commonly used in POST requests).
3. **204 No Content:** The request was successful, but there's no content to return (often used in DELETE requests).
4. **400 Bad Request:** The server cannot process the request due to client error (e.g., malformed request syntax).
5. **401 Unauthorized:** Authentication is required and has failed or has not been provided.
6. **403 Forbidden:** The server understands the request but refuses to authorize it.
7. **404 Not Found:** The requested resource could not be found on the server.
8. **500 Internal Server Error:** A generic error message indicating that the server encountered an unexpected condition.

application files types

application.properties

(.) control it

application.yml

(:) control it



@ControllerAdvice



in Spring is a powerful annotation used to handle exceptions globally across your application. It allows you to define a single class that can handle exceptions thrown by multiple controllers, providing a centralized way to manage error handling and response formatting.

HTTP request - headers

In an HTTP request, headers are key-value pairs sent between the client (such as a web browser or Postman) and the server. They provide essential information about the request or the client itself. Here's a breakdown of what headers are and their purposes:

Internationalization

Internationalization (**i18n**) in Spring allows you to create applications that can adapt to different languages and cultures.