Validation

- dasturlash.uz

Validation

- **Hibernate validator**
- **Javax** bean **validation**

- **Both of them are implemented JSP 303 specification**

- dasturlash.uz

Javax bean validation

- The Java Bean **Validation's @Valid** constraint annotation makes sure that when an object is **validated**, the **validation** recurses to all fields that are annotated with **@Valid** .
- This makes it really easy to perform the usually complex task of **validating** entire object graphs.

- ```
  <dependency>
      <groupId>javax.validation</groupId>
      <artifactId>validation-api</artifactId>
      <version>2.0.1.Final</version>
  </dependency>

  <dependency>
      <groupId>org.hibernate.validator</groupId>
      <artifactId>hibernate-validator</artifactId>
      <version>6.0.13.Final</version>
  </dependency>
  ```

- dasturlash.uz
  ```

Spring boot validation

- Boyagi 2ta libni o’rniga buni ishlatsa bo’ladi.

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

- dasturlash.uz

Validation annotations

- ***@NotNull*** validates that the annotated property value is not *null*.

  ```
  @NotNull(message = "Name cannot be null")
    private String name;
  ```

- ***@AssertTrue*** validates that the annotated property value is *true*.

  ```
  @AssertTrue private
  boolean working;
  ```

- ***@Size*** validates that the annotated property value has a size between the attributes *min* and *max*; can be applied to *String*, *Collection*, *Map*, and array properties.

  ```
  @Size(min = 10, max = 200, message = "About Me must be between 10 and 200
  characters")
  private String aboutMe;
  ```

- dasturlash.uz

Validation annotations

- ***@Min*** validates that the annotated property has a value no smaller than the *value* attribute.
- ***@Max*** validates that the annotated property has a value no larger than the *value* attribute.

```
@Min(value = 18, message = "Age should not be less than 18")
@Max(value = 150, message = "Age should not be greater than 150")
 private int age;
```

- ***@Email*** validates that the annotated property is a valid email address.

```
@Email(message = "Email should be valid")
private String email;
```

Validation annotations

- ***@NotEmpty*** validates that the property is not null or empty; can be applied to *String*, *Collection*, *Map* or *Array* values.

  @NotEmpty(message = "Name field must has a value")
  private String name;

  NotEmpty = NotNull + size/length > 0

- ***@NotBlank*** can be applied only to text values and validates that the property is not null or whitespace.

  @NotBlank(message = "Field must have some value")
  private String description;

  NotBlank = NotNull + .trim().length()

-

Validation annotations

- ***@Positive*** and ***@PositiveOrZero*** apply to numeric values and validate that they are strictly positive, or positive including 0.

- @Negative and @NegativeOrZero apply to numeric values and validate that they are strictly negative, or negative including 0.
- ***@Past*** and ***@PastOrPresent*** validate that a date value is in the past or the past including the present; can be applied to date types including those added in Java 8.
- ***@Future*** *and* ***@FutureOrPresent*** validate that a date value is in the future, or in the future including the present.

- dasturlash.uz

Validation annotations

- @DecimalMax The value of the field or property must be a decimal value lower than or equal to the number in the value element.
- @DecimalMin The value of the field or property must be a decimal value greater than or equal to the number in the value element.

```
@DecimalMax(value = "99999.999", message = "The decimal value can not be more than 99999.999 ")
@DecimalMin(value = "1.00", message = "The decimal value can not be less than 1.00 digit ")
 private float amount = 0f;
```

How to send message in bad request

- dasturlash.uz

ResponseEntityExceptionHandler 1

- Exception while binding

```
@ControllerAdvice
public class CustomGlobalExceptionHandler extends ResponseEntityExceptionHandler {
 @Override
    protected ResponseEntity<Object> handleMethodArgumentNotValid( MethodArgumentNotValidException ex,
                                                                   HttpHeaders headers,
                                                                   HttpStatus status, WebRequest request) {


     // write logic there
    }

}
```

- dasturlash.uz

ResponseEntityExceptionHandler 2

- Response message

```
Map<String, Object> body = new LinkedHashMap<>();
body.put("timestamp", new Date());
body.put("status", status.value());

List<String> errors = new LinkedList<>();
for (FieldError error : ex.getBindingResult().getFieldErrors()) {
    errors.add(error.getDefaultMessage());
}

body.put("errors", errors);
return new ResponseEntity<>(body, headers, status);
```