Clear illustration : https://www.javaguides.net/2021/03/validation-in-spring-boot-rest-api-with-hibernate-validator.html

Hibernate Validator, All these validator can be used in Pojo class which is responsible for data base table createin in hibernate used .

* **@NotNull** validates that the annotated property value is not null.
* **@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.
* **@Min** validates that the annotated property has a value not smaller than the value attribute.
* **@Max** validates that the annotated property has a value no larger than the value attribute.
* **@Email**validates that the annotated property is a valid email address.
* **@NotEmpty** validates that the property is not null or empty; can be applied to String, Collection, Map, or Array values.
* **@NotBlank** can be applied only to text values and validates that the property is not null or whitespace
* **@Pattern**

Eg.

@Table(name = "users")

@Entity

public class UserDTO {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private long id;

@Column(name = "name", nullable = false)

// user name should not be null or empty

// user name should have at least 2 characters

@NotEmpty

@Size(min = 2, message = "user name should have at least 2 characters")

private String name;

// email should be a valid email format

@Email (message=”Email id is not in format, please check”)

String email

We use in controller method parameter where this UserDTO act as a parameter with @valid annotation

@RestController

@RequestMapping("/api/")

public class UserController {

@Autowired

private UserService userService;

@PostMapping("users")

public ResponseEntity<User> createUser(@Valid @RequestBody UserDTO userDTO){

User savedUser = userService.createUser(user);

return new ResponseEntity<User>(savedUser, HttpStatus.CREATED);

}

}

Note that we enable validation on Spring Rest Controller by adding **@Valid** annotation in addition to **@RequestBody.**