Contents

Link github: https://github.com/ngatdo3003/validation-exceptionHandler-example

Validation Entity – Exception Handler

1. Validation Annotation

ANNOTATION	USAGE
@AssertFalse	The annotated element must be false.
@AssertTrue	The annotated element must be true.
@DecimalMax	The annotated element must be a number whose value must be lower or equal to the specified maximum.
@DecimalMin	The annotated element must be a number whose value must be higher or equal to the specified minimum.
@Future	The annotated element must be an instant, date or time in the future.
@Max	The annotated element must be a number whose value must be lower or equal to the specified maximum.
@Min	The annotated element must be a number whose value must be higher or equal to the specified minimum.
@Negative	The annotated element must be a strictly negative number.
@NotBlank	The annotated element must not be null and must contain at least one non-whitespace character.
@NotEmpty	The annotated element must not be null nor empty.
@NotNull	The annotated element must not be null.
@Null	The annotated element must be null.
@Pattern	The annotated CharSequence must match the specified regular expression.
@Positive	The annotated element must be a strictly positive number.
@Size	The annotated element size must be between the specified boundaries (included).

2. Cấu hình file pom.xml

```xml
<properties>
    <java.version>1.8</java.version>
    <lombok.version>1.18.12</lombok.version>
</properties>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
        <optional>true</optional>
    </dependency>
    <!-- https://mvnrepository.com/artifact/javax.validation/validation-api -->
    <dependency>
        <groupId>javax.validation</groupId>
        <artifactId>validation-api</artifactId>
        <version>2.0.1.Final</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.10.2</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.12</version>
        <scope>provided</scope>
    </dependency>
</dependencies>
<dependencyManagement>
    <dependencies>
```

```xml
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>${spring-cloud.version}</version>
        <type>pom</type>
        <scope>import</scope>
    </dependency>
    </dependencies>
</dependencyManagement>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <!-- put your configurations here -->
                <source>1.8</source>
                <target>1.8</target>
                <annotationProcessorPaths>
                    <path>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
                        <version>${lombok.version}</version>
                    </path>
                </annotationProcessorPaths>
            </configuration>
        </plugin>

    </plugins>
</build>
```

3. Cấu hình file application.properties

```properties
# ==============================
# DATABASE
# ==============================

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/test2
spring.datasource.username=root
spring.datasource.password=123456


# ==============================
# JPA / HIBERNATE
# ==============================

#spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

4. Entity

Tạo class UserEnt cùng với các annotation của JPA và Validation

```java
@Entity
@Data // lombok annotation for auto-generate getter/setter/constructor
@Table(name = "users", schema = "test2", catalog = "")
public class UserEnt implements Serializable {
    @Id
    @Column(name = "id", updatable = false, nullable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Basic
    @Column(name = "userid")
    @NotBlank
    private String userid;

    @Basic
    @Column(name = "password")
    @Size(min = 6, max = 20)
    private String password;

    @Basic
    @Column(name = "email")
    @Email(message = "email should be a valid email")
    private String email;

    @Basic
    @Column(name = "phone")
    @Pattern(regexp = "[0-9]{10}")
    private String phone;

    @Basic
    @Column(name = "firstName")
    @NotBlank(message = "first name must not be blank")
    private String firstName;

    @Basic
    @Column(name = "lastName")
    @NotBlank(message = "last name must not be blank")
    private String lastName;

    @Basic
    @Column(name = "expired")
    private int expired;

    @Basic
    @Column(name = "disabled")
    private int disabled;

    @Basic
    @Column(name = "age")
    @Min(13)
    @Max(99)
    private int age;

}
```

5. Repository

Kế thừa lớp JpaRepository gồm các thao tác với csdl cơ bản:

```java
public interface UserRepository extends JpaRepository<UserEnt, Long> {
}
```

6. Customized Exception

a. RecordNotFoundException

```java
@ResponseStatus(HttpStatus.NOT_FOUND)
public class RecordNotFoundException extends RuntimeException {
    public RecordNotFoundException(String exception) {
        super(exception);
    }
}
```

b. InvalidUseridException

```java
@ResponseStatus(HttpStatus.NOT_ACCEPTABLE)
public class InvalidUserIDException extends RuntimeException {
    public InvalidUserIDException(String exception) {
        super(exception);
    }
}
```

c. CustomExeptionHandler

```java
@SuppressWarnings({"unchecked","rawtypes"})
@ControllerAdvice
public class CustomExceptionHandler extends ResponseEntityExceptionHandler
{
    @ExceptionHandler(Exception.class)
    public final ResponseEntity<Object> handleAllExceptions(Exception ex,
WebRequest request) {
        List<String> details = new ArrayList<>();
        details.add(ex.getLocalizedMessage());
        ErrorResponse error = new ErrorResponse("Server Error", details);
        return new ResponseEntity(error, HttpStatus.INTERNAL_SERVER_ERROR);
    }

    @ExceptionHandler(RecordNotFoundException.class)
    public final ResponseEntity<Object>
handleUserNotFoundException(RecordNotFoundException ex, WebRequest request) {
        List<String> details = new ArrayList<>();
        details.add(ex.getLocalizedMessage());
        ErrorResponse error = new ErrorResponse("Record Not Found", details);
        return new ResponseEntity(error, HttpStatus.NOT_FOUND);
    }

    @ExceptionHandler(InvalidUserIDException.class)
    public final ResponseEntity<Object>
handleInvalidUserIDException(InvalidUserIDException ex, WebRequest request) {
        List<String> details = new ArrayList<>();
        details.add(ex.getLocalizedMessage());
```

```java
        ErrorResponse error = new ErrorResponse("Invalid userID ", details);
        return new ResponseEntity(error, HttpStatus.NOT_ACCEPTABLE);
    }


    @Override
    protected ResponseEntity<Object>
handleMethodArgumentNotValid(MethodArgumentNotValidException ex, HttpHeaders
headers, HttpStatus status, WebRequest request) {
        List<String> details = new ArrayList<>();
        for(ObjectError error : ex.getBindingResult().getAllErrors()) {
            details.add(error.getDefaultMessage());
        }
        ErrorResponse error = new ErrorResponse("Validation Failed",
details);
        return new ResponseEntity(error, HttpStatus.BAD_REQUEST);
    }
}
```

7. ValidatingService

```java
/**
 * Service validate dữ liệu dựa trên annotation @NotNull @NotBlank @Size @Min
@Max...
 */
@Service("ValidatingService")
@Validated
public class ValidatingService {
    /**
     * Hàm validate theo các Annotation
     * @param input
     */
    public void validate(@Valid Object input){
    }

    /**
     * Hàm validate theo Annotation và thêm điều kiện là userID phải bắt đầu
bằng ABC
     * @param user
     * @return
     */
    public void customizedValidate(@Valid UserEnt user) throws
InvalidUserIDException {
        if(!(user.getUserid()).startsWith("ABC")) throw new
InvalidUserIDException("UserID: " + user.getUserid() + " is invalid");
    }
}
```

8. Controller

```java
@RestController
public class UserController {

    @Autowired
    UserRepository userRepository;

    @Autowired
```

```java
    ValidatingService validatingService;

    /**
     * Service thêm mới user
     * validate dữ liệu (chỉ validate dựa trên các annotation trong Entity)
ngay từ khi convert từ JSON sang Ent
     * @param user
     * @return
     */
    @PostMapping(value = "/user/add/1")
    public ResponseEntity<UserEnt> addUser1(@RequestBody @Valid UserEnt user)
    {
        user.setExpired(1);
        user.setDisabled(0);
        userRepository.save(user);
        return new ResponseEntity<UserEnt>(user, HttpStatus.OK);
    }

    /**
     * Service thêm mới user
     * Không validate dữ liệu khi convert từ JSON sang Ent
     * Validate dữ liệu sau đó dùng validatingService(validate dựa trên
annotation trong Entity và customized validation
     * @param user
     * @return
     */

    @PostMapping(value = "/user/add/2")
    public ResponseEntity<UserEnt> addUser2(@RequestBody  UserEnt user)
throws InvalidUserIDException {
        user.setExpired(1);
        user.setDisabled(0);
        validatingService.customizedValidate(user);
        userRepository.save(user);
        return new ResponseEntity<UserEnt>(user, HttpStatus.OK);
    }

    /**
     * Service lấy thông tin User theo id
     * @param id
     * @return
     */
    @GetMapping(value = "/user/{id}")
    public ResponseEntity<UserEnt> findById (@PathVariable("id") Long id)
    {
        Optional<UserEnt> user = userRepository.findById(id);
        if(!user.isPresent()) {
            throw new RecordNotFoundException("Invalid user id : " + id);
        }
        return new ResponseEntity<UserEnt>(user.get(), HttpStatus.OK);
    }
}
```

9. **Demo**
- Gọi đến web service add1 với các field không hợp lệ, ta được kết quả:

```
POST        ▼    http://localhost:8080/user/add/1

1 ▼ {
2       "userid": "A1234",
3       "password":"1234",
4       "email":"abcgmail.com",
5       "phone":"1213",
6       "firstName":"Do",
7       "lastName":"    ",
8       "age":12
9   }
```

Body   Cookies   Headers (4)   Test Results   Status: 400 Bad Request   Time: 23 ms

Pretty   Raw   Preview   Visualize   JSON ▼

```
1  {
2      "message": "Validation Failed",
3      "details": [
4          "email should be a valid email",
5          "must match \"[0-9]{10}\"",
6          "last name must not be blank",
7          "size must be between 6 and 20",
8          "must be greater than or equal to 13"
9      ]
10 }
```

- Gọi đến web service add2 với các filed không hợp lệ, ta sẽ được báo lỗi của các filed được validate theo annotation trước


```
POST        ▼    http://localhost:8080/user/add/2                    Send  ▼   Save ▼

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings        Cookies  Code

● none  ● form-data  ● x-www-form-urlencoded  ● raw  ● binary  ● GraphQL   JSON ▼           Beautify

1 ▼ {
2       "userid": "A1234",
3       "password":"1234",
4       "email":"abcgmail.com",
5       "phone":"1213",
6       "firstName":"Do",
7       "lastName":"   ",
8       "age":12
9   }
```

Body   Cookies   Headers (4)   Test Results   Status: 500 Internal Server Error   Time: 1052 ms   Size: 506 B   Save Response ▼

Pretty   Raw   Preview   Visualize   JSON ▼

```
1  {
2      "message": "Server Error",
3      "details": [
4          "customizedValidate.user.phone: must match \"[0-9]{10}\", customizedValidate.user.lastName: last name must not be blank,
              customizedValidate.user.email: email should be a valid email, customizedValidate.user.password: size must be between 6 and 20,
              customizedValidate.user.age: must be greater than or equal to 13"
5      ]
6  }
```

- Sau khi nhập dữ liệu hợp lệ, chỉ trừ field userid, ta sẽ thấy thông báo lỗi:

```
POST ▼   http://localhost:8080/user/add/2                              Send ▼

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings

● none   ● form-data   ● x-www-form-urlencoded   ● raw   ● binary   ● GraphQL   JSON ▼

1 ▼ {
2      "userid": "A1234",
3      "password":"123456",
4      "email":"abc@gmail.com",
5      "phone":"1234567890",
6      "firstName":"Do",
7      "lastName":"Ngat",
8      "age":21
9   }

Body   Cookies   Headers (4)   Test Results        Status: 500 Internal Server Error   Time: 258 ms   Size: 221 B   Save

Pretty   Raw   Preview   Visualize   JSON ▼

1  {
2      "message": "Server Error",
3      "details": [
4          "Userid must start with ABC"
5      ]
6  }
```

- Nhập đúng dữ liệu, gửi request thành công:

```
GET ▼   http://localhost:8080/user/2                                  Send ▼

Params   Authorization   Headers (7)   Body   Pre-request Script   Tests   Settings

● none   ● form-data   ● x-www-form-urlencoded   ● raw   ● binary   ● GraphQL

                    This request does not have a body

Body   Cookies   Headers (5)   Test Results             Status: 200 OK   Time: 83 ms   Size: 325 B   Save

Pretty   Raw   Preview   Visualize   JSON ▼

1  {
2      "id": 2,
3      "userid": "ABC1234",
4      "password": "123456",
5      "email": "abc@gmail.com",
6      "phone": "1234567890",
7      "firstName": "Do",
8      "lastName": "Ngat",
9      "expired": 1,
10     "disabled": 0,
11     "age": 21
12 }
```

- Gọi web service lấy thông tin một user không có trong DB

10

| GET ▼ | http://localhost:8080/user/3 | Send ▼ |

Params   Authorization   Headers (7)   **Body**   Pre-request Script   Tests   Settings

● none   ○ form-data   ○ x-www-form-urlencoded   ○ raw   ○ binary   ○ GraphQL

This request does not have a body

Body   Cookies   Headers (5)   Test Results   Status: 404 Not Found   Time: 585 ms   Size: 235 B   Sav

Pretty   Raw   Preview   Visualize   JSON ▼

```
1  {
2      "message": "Record Not Found",
3      "details": [
4          "Invalid user id : 3"
5      ]
6  }
```