

# Bài 18

# Validation

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT

# Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài “Spring Data Repository”  
Tóm tắt lại các phần đã học từ bài “Spring Data Repository”

- Cấu hình được Validation trong ứng dụng Spring MVC
- Validate đối tượng bằng annotation
- Custom Validator

# Thảo luận

Giới thiệu về Validation

# Khái niệm Validation

---

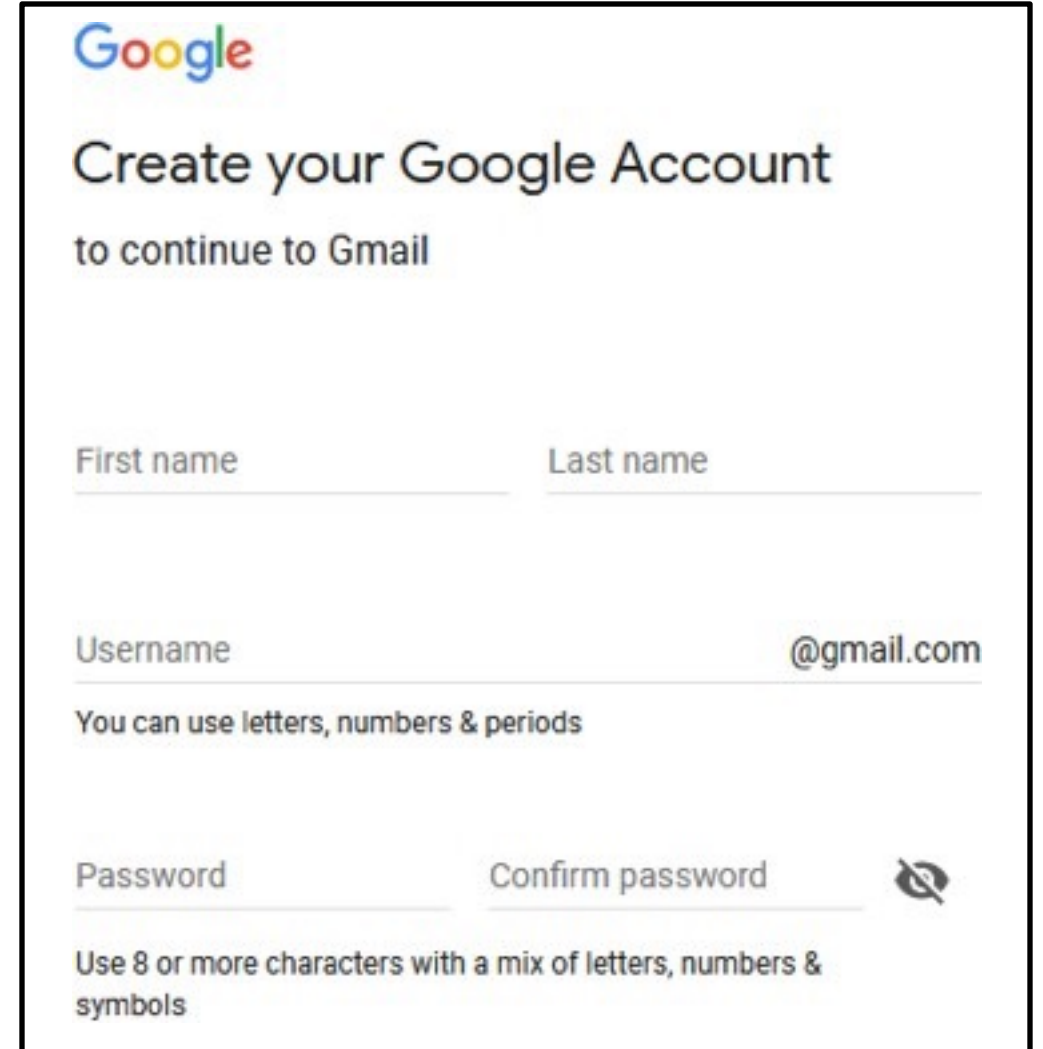
- Validation thực hiện đánh giá/xác minh tính hợp lệ của dữ liệu đầu vào.

# Ví dụ

Ví dụ: Form đăng ký tài khoản Google

First name, Last name, Username  
không để trống

Password có độ dài >8 ký tự,  
có chứa số, chữ, ký tự đặc biệt



The image shows a screenshot of the Google Account creation page. At the top is the Google logo. Below it, the text reads "Create your Google Account" and "to continue to Gmail". There are four input fields: "First name", "Last name", "Username", and "Password". The "Username" field has a placeholder "@gmail.com" and a note below it: "You can use letters, numbers & periods". The "Password" field has a "Confirm password" field next to it and a note below: "Use 8 or more characters with a mix of letters, numbers & symbols". There is also a small icon of a crossed-out eye next to the password fields.

Google

Create your Google Account  
to continue to Gmail

First name Last name

Username @gmail.com

You can use letters, numbers & periods

Password Confirm password

Use 8 or more characters with a mix of letters, numbers & symbols

- Các tầng Validation dữ liệu



- Spring hỗ trợ validation ở tầng business

# Thảo luận

Spring Validation



# Dependency Validate trong Spring

---

- Dependency

```
compile group: 'javax.validation', name: 'validation-api', version: '2.0.1.Final'  
compile group: 'org.hibernate', name: 'hibernate-validator', version: '6.0.10.Final'
```

# Validate chuỗi

Annotation	Ví dụ
@NotEmpty	@NotEmpty private String firstName;
@NotBlank	@NotBlank private String firstName;
@Email	@Email private String email;
@Pattern	@Pattern(regexp="^[A-Za-z]*\$") private String lastName; @Pattern(regexp="(^\$ [0-9]{10})") private String number;
@Size	@Size(min=1,max=20) private String lastName;

# Validate số

Annotation	Ví dụ
@NumberFormat	@NumberFormat(pattern="#.##") private Double weight;
@Min	@Min(18) Private int age;
@Max	@Max(200) private int weigth

# Validate Date

---

Annotation	Ví dụ
@DateTimeFormat	@DateTimeFormat(pattern = "MM/dd/yyyy") private Date dob;

# Controller Validate

---

- @Validated/@Valid
- BindingResult

Ví dụ:

```
@PostMapping("/")
public String checkValidation (@Validated @ModelAttribute("user") User user,
BindingResult bindingResult){
    if (bindingResult.hasFieldErrors()){
        return "index";
    }
    return "result";
}
```

# Tùy biến thông báo trong Annotation

---

message = "" tùy chỉnh thông báo mặc định

Ví dụ:

```
@NotEmpty(message = "Name not empty")  
@Size(min = 2, max=30, message = "Length from 2 to 30")
```

# Demo

Validate bằng anotation

# Thảo luận

Custom Validator



# Custom Validator Model

- Cung cấp Interface Validation và hỗ trợ xác minh đối tượng

`@Override`

```
public boolean supports(Class<?> clazz) {  
    return PhoneNumber.class.isAssignableFrom(clazz);  
}
```

`@Override`

```
public void validate(Object target, Errors errors) {  
  
    PhoneNumber phoneNumber = (PhoneNumber) target;  
    String number = phoneNumber.getNumber();  
    ValidationUtils.rejectIfEmpty(errors, "number", "number.empty");  
}
```

Trong đó:

- PhoneNumber là class được validate
  - validate(Object target, Errors errors) là phương thức xác minh đối tượng target
- Và trả về lỗi nếu có trong errors
- Number trường trả về lỗi với number.empty chứa lỗi

# Validator Controller

---

- Trong Controller gọi phương thức validate

```
new PhoneNumber().validate(phoneNumber, bindingResult);  
if (bindingResult.hasFieldErrors()){  
    ...;  
}  
else {  
    ...;  
}
```

# Cấu hình ValidationMessages

- Cấu hình file ValidationMessages.properties trong resousce chứa lỗi

@Bean

```
public MessageSource messageSource() {  
    ResourceBundleMessageSource messageSource =  
new ResourceBundleMessageSource();  
    messageSource.setBasenames("ValidationMessages");  
    return messageSource;  
}
```

- Ví dụ: trong ValidationMessages.properties chứa  
number.empty = phonemunber not empty.

# Hiển thị lỗi tại view

---

- Thymeleaf: thuộc tính th:errors

Ví dụ:

```
<td th:if="${#fields.hasErrors('name')}" th:errors="*{name}">Name Error</td>
```

- JPS: thuộc tính path trong thẻ form:errors

Ví dụ:

```
<form:errors path="name"></form:errors>
```

# Demo

Custom Validator

# Tóm tắt bài học

---

- Validation xác minh tính hợp lệ của đối tượng
- Các cấu hình validation trong Spring MVC
- Cách validation một Model bằng Annotation
- Custom Validator

# Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: Session and Cookie