FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

FORM AND DATABINDING

GIẢNG VIÊN:

www.poly.edu.vn

FPT Education
FPT POLYTECHNIC

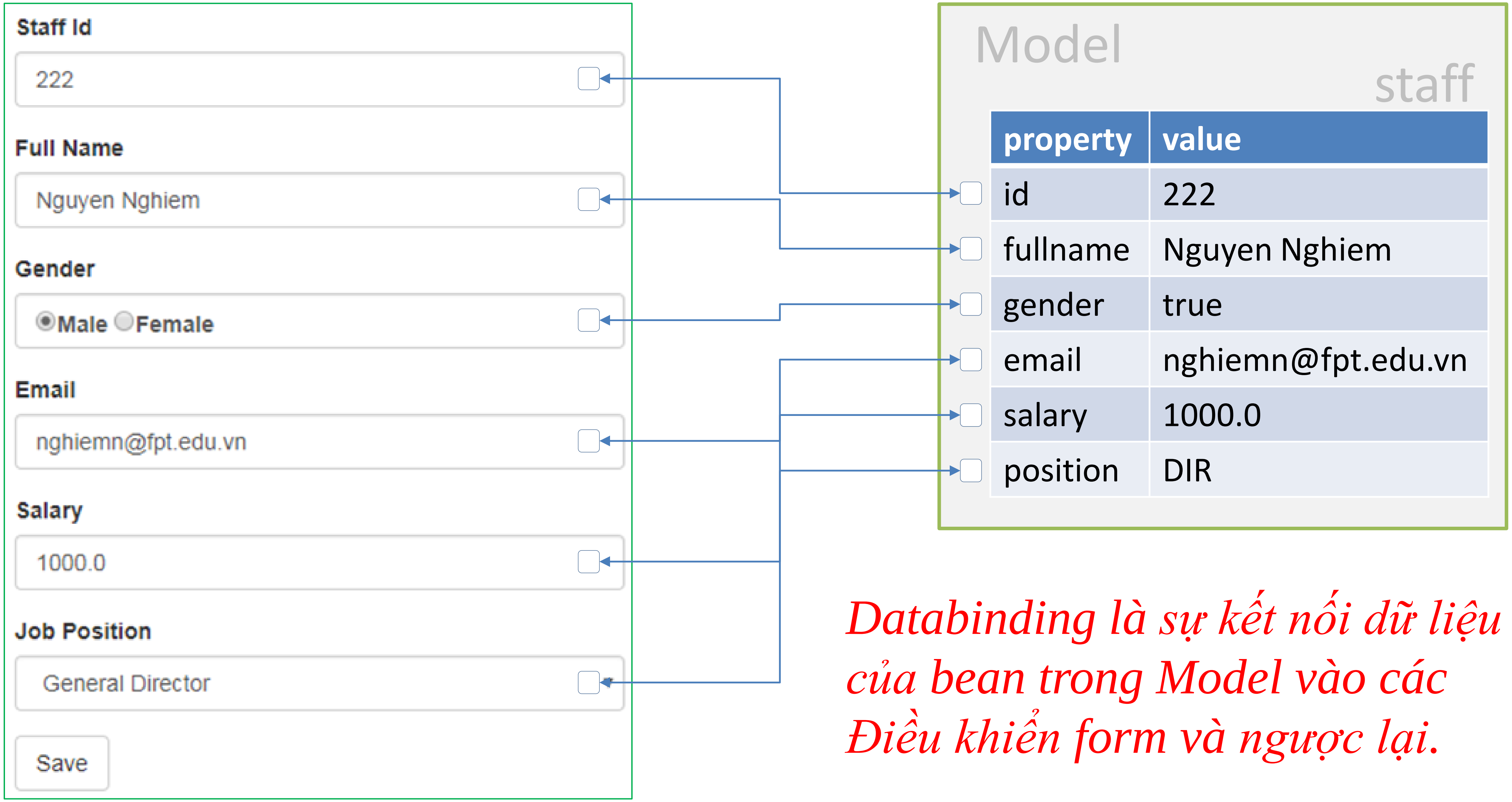
Agenda

- Form & Databinding
  - Understanding Databinding
  - Spring Form and Databinding
  - Upload file
- Validation
  - Understanding Validation
  - Validation bean
  - Validation Annotations
  - Validation Controller
  - Displaying error messages
  - Customizing error messages

Databinding

FPT Education
FPT POLYTECHNIC

WHAT IS DATABINDING?



Staff Id	222
Full Name	Nguyen Nghiem
Gender	● Male ○ Female
Email	nghiemn@fpt.edu.vn
Salary	1000.0
Job Position	General Director

Save

Model

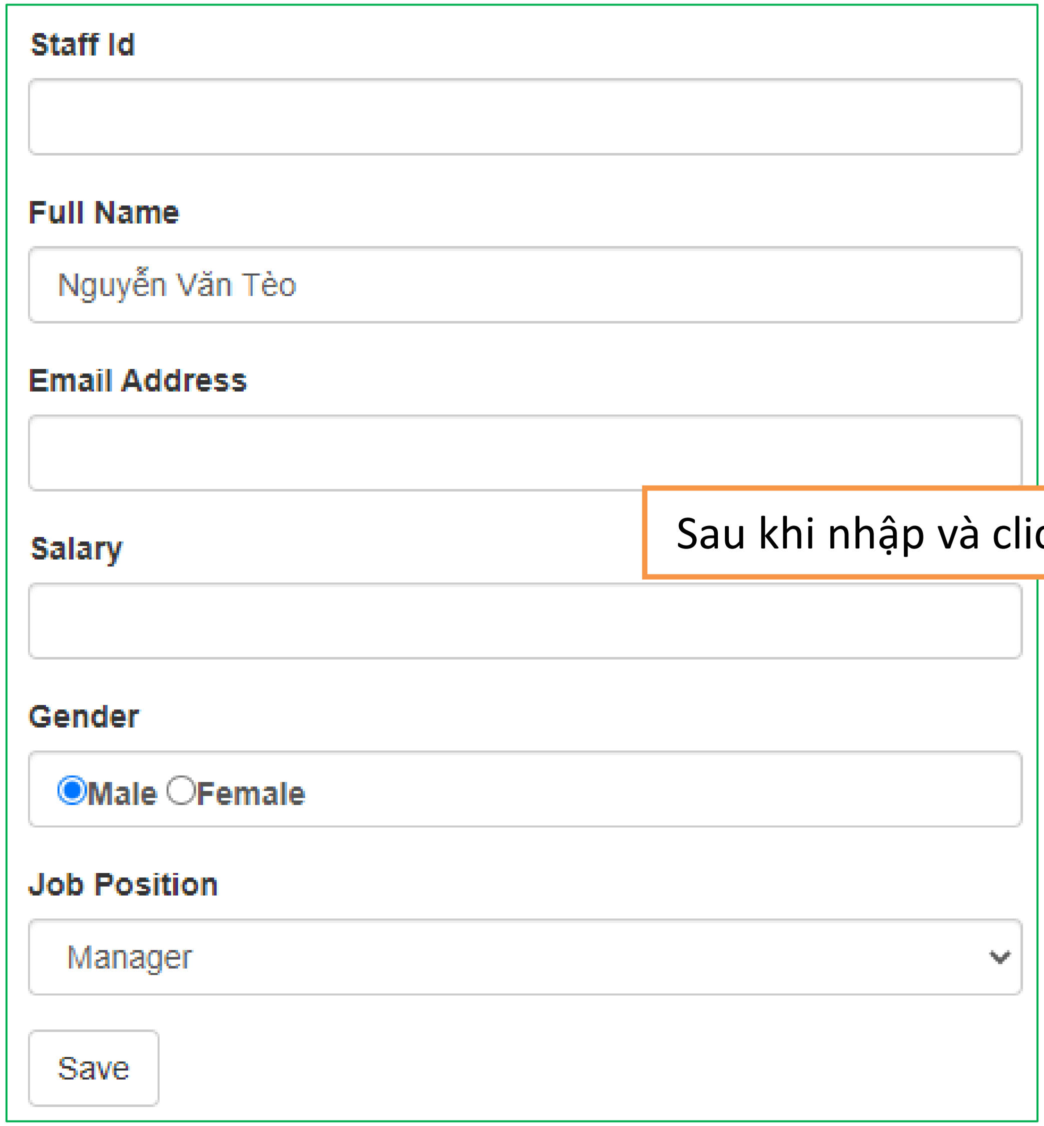
staff

property	value
id	222
fullname	Nguyen Nghiem
gender	true
email	nghiemn@fpt.edu.vn
salary	1000.0
position	DIR

Databinding là sự kết nối dữ liệu của bean trong Model vào các Điều khiển form và ngược lại.

DATABINDING

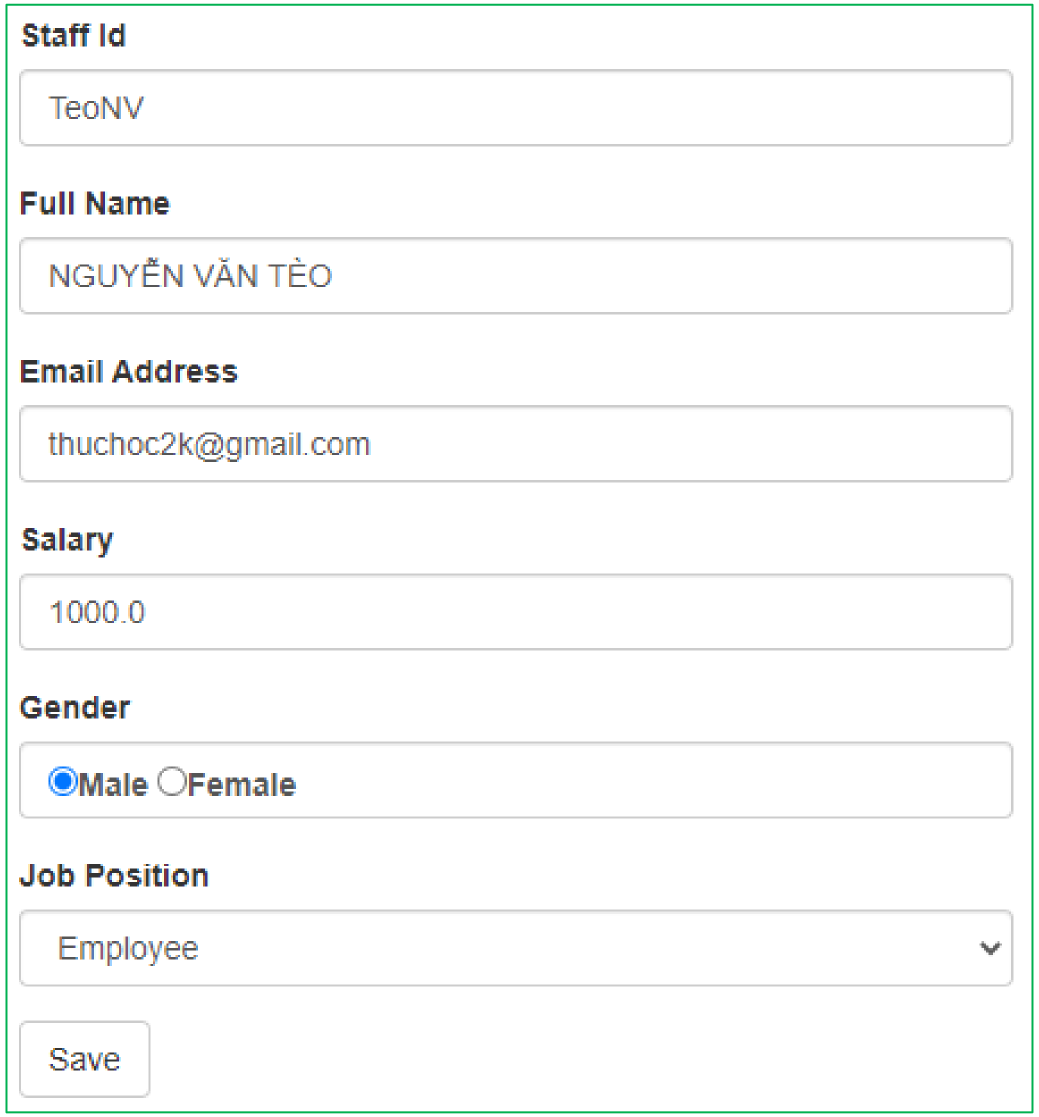
CASE STUDY #1



Staff Id	
Full Name	Nguyễn Văn Tèo
Email Address	
Salary	
Gender	[✓ Male] Female
Job Position	Manager

Save

Sau khi nhập và click [Save]



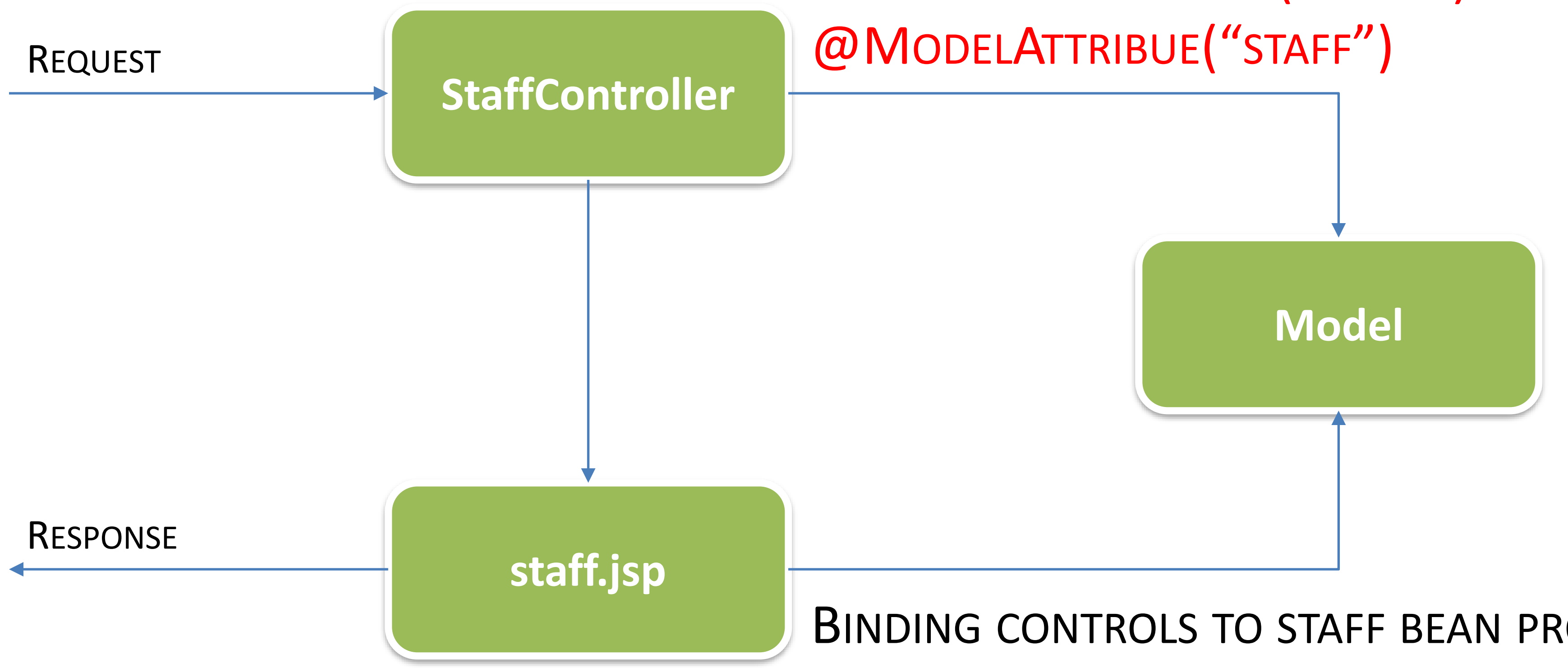
Staff Id	TeoNV
Full Name	NGUYỄN VĂN TÈO
Email Address	thuchoc2k@gmail.com
Salary	1000.0
Gender	[✓ Male] Female
Job Position	Employee

Save

KHỞI TẠO FORM

DUY TRÌ TRẠNG THÁI FORM

Case Study Solution



Add staff bean to Model

Model.addAttribute("staff")

@ModelAttribue("staff")

Binding controls to staff bean properties

```
<form:form modelAttribute="staff">
    <form:input path="fullname"/>
</form:form>
```

Staff Bean Class

```
public class Staff {
  String id;
  String fullname;
  String email;
  Double salary;
  Boolean gender;
  String position;
  getters/setters
}
```

Spring Form

SPRING FORM

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<form:form action="staff/save" modelAttribute="staff">
        <form:input path="id"/>
        <form:input path="fullname"/>
        <form:input path="email"/>
        <form:input path="salary"/>
        <form:radiobutton path="gender" value="true" label="Male"/>
        <form:radiobutton path="gender" value="false" label="Female"/>
        <form:select path="position">
                <form:option value="CEO">Chief Executive Officer</form:option>
                <form:option value="DIR">Director</form:option>
                <form:option value="MAN">Manager</form:option>
                <form:option value="EMP">Employee</form:option>
        </form:select>
        <button>Save</button>
</form:form>
```

Bộ thư viện thẻ Spring Form giúp thực hiện databinding một cách dễ dàng

CONTROLLER

```
@GetMapping("staff/index")
public String index(Model model) {
        Staff staff = new Staff();
        staff.setFullname("Nguyễn Văn Tèo");
        staff.setGender(true);
        staff.setPosition("MAN");
        model.addAttribute("staff", staff);
        return "staff";
}
@PostMapping("staff/save")
public String save(@ModelAttribute("staff") Staff staff) {
        staff.setFullname(staff.getFullname().toUpperCase());
        return "staff";
}
```

Model

Spring Form Tags

Generate HTML Tags

1. <form:form>	1. <form>
2. <form:input/>	2. <input type="text"/>
3. <form:textarea/>	3. <textarea/>
4. <form:checkbox/>	4. <input type='checkbox'/>
5. <form:radiobutton/>	5. <input type='radio'/>
6. <form:hidden/>	6. <input type='hidden'/>
7. <form:password/>	7. <input type='password'/>
8. <form:button/>	8. <button/>
9. <form:select/>	9. <select/>

FPT Education
FPT POLYTECHNIC

Spring Form Tags – List Controls

```
<select>
<option value="{value1}">{label1}</option>
<option value="{value2}">{label2}</option>
...
</select>
```

9. <form:select/>

10. <form:radiobuttons/>

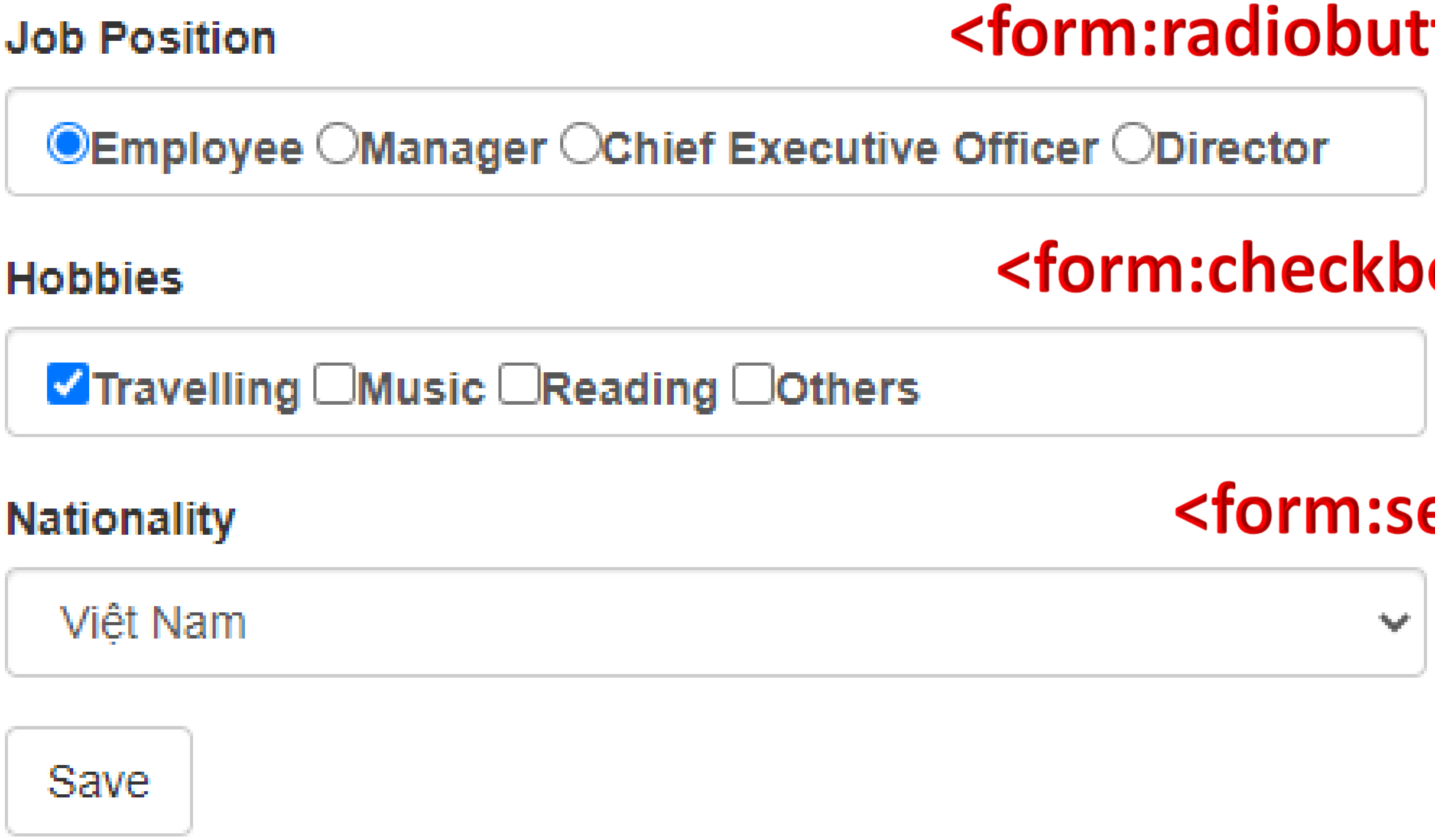
```
<input type="radido" value="{value1}"><label>{label1}</label>
<input type="radido" value="{value2}"><label>{label2}</label>
...
```

```
<input type="checkbox" value="{value1}"><label>{label1}</label>
<input type="checkbox" value="{value2}"><label>{label2}</label>
...
```

11. <form:checkboxes/>

Data filling

CASE STUDY #2



Job Position

● Employee ○ Manager ○ Chief Executive Officer ○ Director

Hobbies

[✓ Travelling] ☐ Music ☐ Reading ☐ Others

Nationality

Việt Nam

<form:radiobuttons items="data">

<form:checkboxes items="data">

<form:select items="data">

Spring Form – Data Filling

```
<form:form action="/staff2/save" modelAttribute="staff2">
    <form:radiobuttons path="position" items="${positions}" delimiter="  "/>
    <form:checkboxes path="hobbies" items="${hobbies}" delimiter="  "/>
    <form:select path="country.id" cssClass="form-control">
        <form:options items="${nationalities}" itemValue="id" itemLabel="name"/>
    </form:select>
    <button class="btn btn-default">Save</button>
</form:form>
```

Datafilling attributes ← **DataSource**

- @items = <datasource>
- @itemValue = <property>
- @itemLabel = <property>
- @delimiter = <element>

- ${positions} <= Map<String, String>
- ${hobbies} <= String[]
- ${nationalities} <= List<Country>

DATA SOURCE FOR FILLING

```
@Controller
public class Staff2Controller {
        @GetMapping("/staff2/index")
        public String index(Model model) {...}
        @PostMapping("/staff2/save")
        public String save(
        @ModelAttribute("staff2") Staff2 staff2) {...}

        @ModelAttribute("positions")
        public Map<String, String> getPositions(){...}

        @ModelAttribute("hobbies")
        public String[] getHobbies(){...}

        @ModelAttribute("nationalities")
        public List<Country> getNationalities(){...}
}
```

```
public class Staff2 {
        String position;
        String[] hobbies;
        Country country;
        getters/setters
}
```

use

```
public class Country {
        String id;
        String name;
        public Country() {}
        public Country(String id, String name) {
                this.id = id;
                this.name = name;
        }
        getters/setters

}
```

FPT Education
FPT POLYTECHNIC

DATA SOURCE FOR FILLING

```java
@GetMapping("/staff2/index")
public String index(Model model) {
	Staff2 staff2 = new Staff2();
	staff2.setPosition("EMP");
	staff2.setHobbies(new String[] {"Travelling"});
	model.addAttribute("staff2", staff2);
	return "staff2";
}
@PostMapping("/staff2/save")
public String save(@ModelAttribute("staff2") Staff2 staff2) {
	return "staff2";
}
```

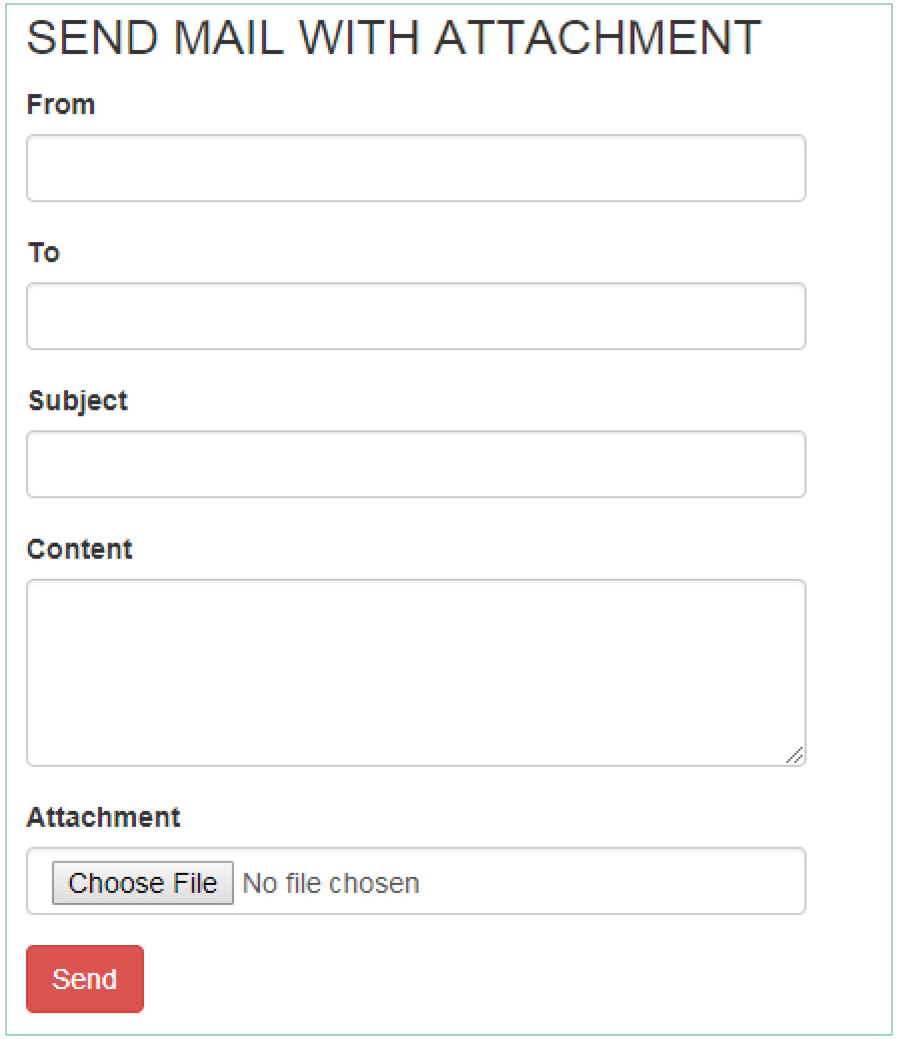
```java
@ModelAttribute("positions")
public Map<String, String> getPositions(){
	Map<String, String> map = new HashMap<>();
	map.put("CEO", "Chief Executive Officer");
	map.put("DIR", "Director");
	map.put("MAN", "Manager");
	map.put("EMP", "Employee");
	return map;
}
```

```java
@ModelAttribute("hobbies")
public String[] getHobbies(){
	String[] hobbies = {"Travelling",
			"Music", "Reading", "Others"};
	return hobbies;
}
```

```java
@ModelAttribute("nationalities")
public List<Country> getNationalities(){
	List<Country> list = new ArrayList<>();
	list.add(new Country("VN", "Việt Nam"));
	list.add(new Country("US", "United States"));
	list.add(new Country("SG", "Singapore"));
	return list;
}
```

Upload file

UPLOAD FILE


SEND MAIL WITH ATTACHMENT

From

To

Subject

Content

Attachment

```
<form action="send" method="POST"
enctype="multipart/form-data">
    <input name="from">
    <input name="to">
    <input name="subject" >
    <textarea name="body"> </textarea>
    <input name="attach" type="file">
    <button>Send</button>
</form>
```

FPT Education
FPT POLYTECHNIC

UPLOADCONTROLLER

```java
public class UploadController {
	@Autowired
	ServletContext app;
	@PostMapping("/upload/form")
	public String form() {
		return "upload/form";
	}
	@PostMapping("/upload/save")
	public String send(@RequestParam("attach") MultipartFile attach)
			throws IllegalStateException, IOException {
		if(!attach.isEmpty()) {
			String filename = attach.getOriginalFilename();
			File file = new File(app.getRealPath("/docs/"+ filename ));
			attach.transferTo(file);
		}
		return "upload/success";
	}
}
```

- MultipartFile
  - isEmpty()
  - getOriginalFilename()
  - transferTo(File)

application.properties

```
spring.servlet.multipart.max-file-size=20MB
spring.servlet.multipart.max-request-size=20MB
```

FPT Education

FPT POLYTECHNIC

cdio™
Conceive Design Implement Operate

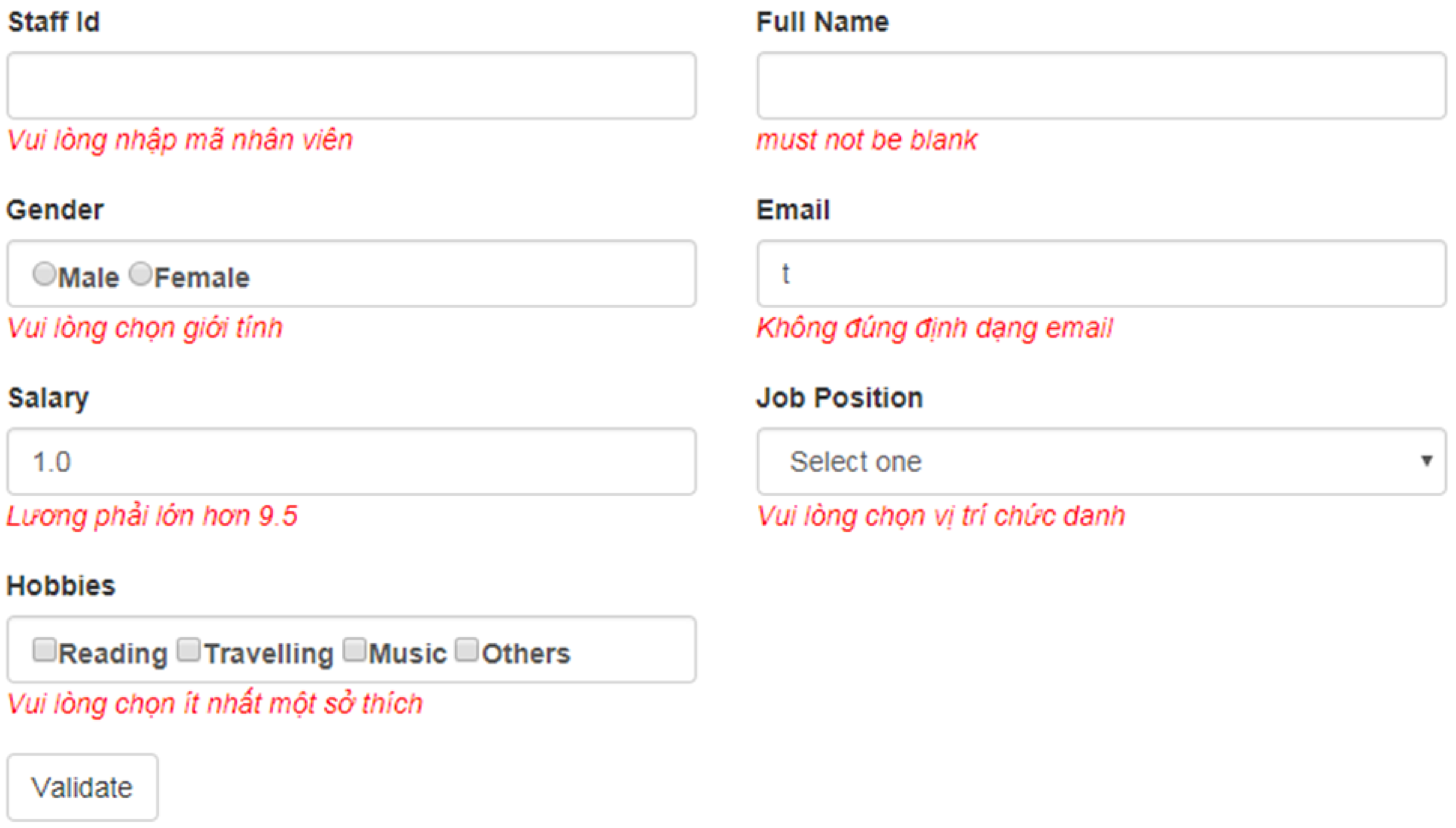
Form Validation

Giảng viên:

www.poly.edu.vn

FPT Education
FPT POLYTECHNIC

Case Study #3


Staff Id

Vui lòng nhập mã nhân viên

Full Name

must not be blank

Gender

( ) Male ( ) Female

Vui lòng chọn giới tính

Email

t

Không đúng định dạng email

Salary

1.0

Lương phải lớn hơn 9.5

Job Position

Select one

Vui lòng chọn vị trí chức danh

Hobbies

☐ Reading ☐ Travelling ☐ Music ☐ Others

Vui lòng chọn ít nhất một sở thích

Validate

FPT Education
FPT POLYTECHNIC

VALIDATION BEAN

```java
public class Staff3 {
    @NotNull
    Integer id;
    @NotBlank
    String fullname;
    @NotNull
    Boolean gender;
    @NotEmpty
    @Email
    String email;
    @NotNull
    @DecimalMin("9.5")
    Double salary;
    @NotEmpty
    String position;
    @NotEmpty
    List<String> hobbies;

    getters/setters
}
```

- @NotNull
  - Không cho phép null
- @NotBlank
  - Không cho để trống (chuỗi)
- @NotEmpty
  - Không cho rỗng (null, chuỗi, map, collection)
- @DecimalMin
  - Giá trị tối thiểu (số)
- @Email
  - Định dạng email

```xml
<dependency>
<groupId>org.hibernate</groupId>
<artifactId>hibernate-validator</artifactId>
<version>5.1.3.Final</version>
</dependency>
```

VALIDATION ANNOTATION

- Null, Empty
  - @NotBlank(), @NotEmpty(), @NotNull(), @Null()
- Number
  - @Negative(), @NegativeOrZero(), @Positive(), @PositiveOrZero()
  - @DecimalMax("5"), @DecimalMin("6.5")
  - @Max(5), @Min(6), @Size(min=0, max=1)
  - @Digits(integer=3, fraction=2)
- Email, Regular Expression
  - @Email(), @Pattern(regexp="")
- Time
  - @Future(), @FutureOrPresent(), @Past(), @PastOrPresent()
- Boolean
  - @AssertFalse(), @AssertTrue()

FPT Education
FPT POLYTECHNIC

VALIDATION CONTROLLER

```
@RequestMapping("save")
public String save(Model model,
        @Valid @ModelAttribute("staff") Staff3 staff,
        BindingResult result) {
    if(result.hasErrors()) {
        model.addAttribute("message",
                "Some fields are not valid. Please fix them!");
    }
    else {
        model.addAttribute("message", "All fields are valid!");
    }
    // view - layout
    model.addAttribute("view", "staff/form3");
    return "bootstrap";
}
```

- @Valid hoặc @Validated: yêu cầu kiểm lỗi bean staff
- BindingResult chứa kết quả kiểm lỗi (khai báo ngay sau bean cần kiểm lỗi)
- result.hasErrors() có lỗi hay không

Display Error Messages

- Hiển thị thông báo lỗi
  - <form:errors path="?" element="?" delimiter="?"/>
    - @path: thuộc tính bean kiểm lỗi hoặc * (tất cả)
    - @element: thẻ bọc thông báo lỗi
    - @delimiter: phân cách các lỗi
- Ví dụ:
  - <form:errors path="*" element="li" delimiter=";"/>
  - <form:errors path="fullname"/>

CUSTOMIZING ERROR MESSAGES

FPT Education
FPT POLYTECHNIC

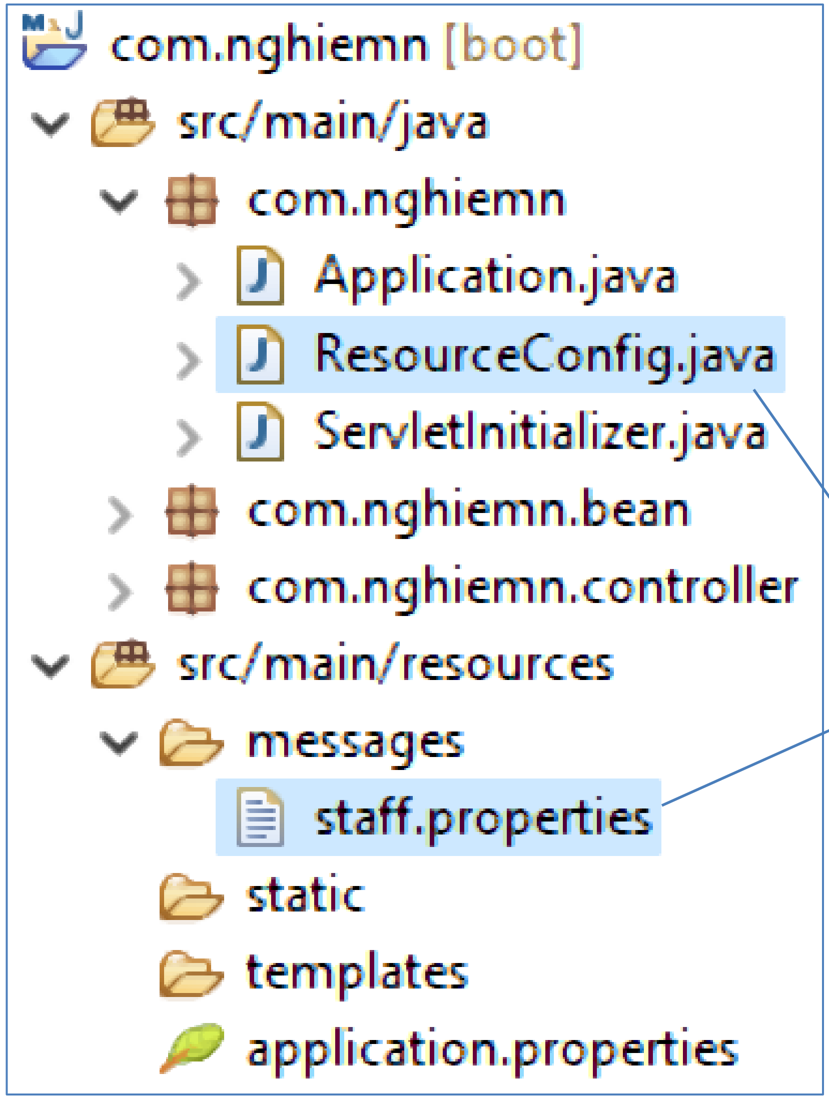
CUSTOMIZING ERROR MESSAGES

- Validation Annotation định nghĩa sẵn thông báo lỗi mặc định.
- Sử dụng thuộc tính message để thay đổi thông báo lỗi
  - @Email(message="Chưa đúng định dạng email")
- Sử dụng file properties để định nghĩa thông báo lỗi tập trung đa ngôn ngữ và cấu hình để nạp tài nguyên thông báo lỗi.

```
NotNull.staff.id=Vui lòng nhập mã nhân viên
NotEmpty.staff.fullname=Vui lòng nhập họ và tên
NotNull.staff.gender=Vui lòng chọn giới tính
NotEmpty.staff.email=Vui lòng nhập email
Email.staff.email=Không đúng định dạng email
typeMismatch=Sai kiểu dữ liệu
```

<annotation>.<bean>.<property>

Error Message Resource Configuration


com.nghiemn [boot]
src/main/java
com.nghiemn
Application.java
ResourceConfig.java
ServletInitializer.java
com.nghiemn.bean
com.nghiemn.controller
src/main/resources
messages
staff.properties
static
templates
application.properties

```
@Configuration
public class ResourceConfig{
        @Bean("messageSource")
        public MessageSource getMessageSource(){
            ReloadableResourceBundleMessageSource ms =
                new ReloadableResourceBundleMessageSource();
            ms.setBasenames("classpath:messages/staff");
            ms.setDefaultEncoding("utf-8");
            return ms;
        }
}
```

Summary

- ☑ Form & Databinding
  - ☑ Understanding Databinding
  - ☑ Spring Form and Databinding
  - ☑ Upload file
- ☑ Validation
  - ☑ Understanding Validation
  - ☑ Validation bean
  - ☑ Validation Annotations
  - ☑ Validation Controller
  - ☑ Displaying error messages
  - ☑ Customizing error messages

FPT Education
FPT POLYTECHNIC
Thank you
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007