



SPRING FORM & VALIDATION

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



OBJECTIVES

- Sử dụng được Spring Form
- Kiểm lỗi Spring Form





AGENDA

- Databinding là gì
- Spring Form và databinding
- Đổ dữ liệu vào list control
- Kiểm lỗi form
- Khai báo thông báo lỗi

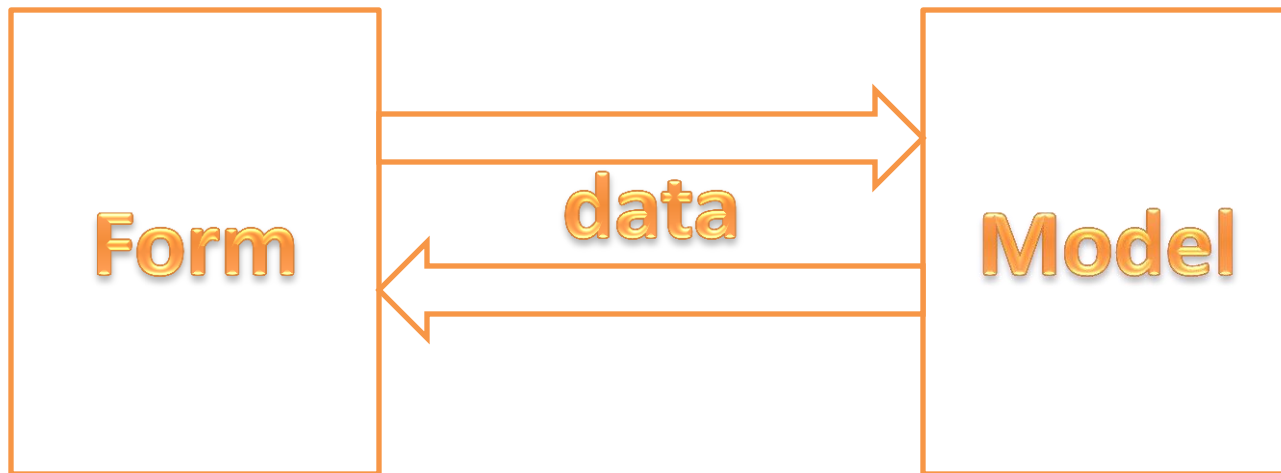


PART 1: SPRING FORM



DATABINDING?

- Databinding là sự kết nối dữ liệu của bean trong Model và các điều khiển trên form





CASE STUDY 1

SIMPLE SPRING FORM

Staff Id

Full Name

Gender

☐ Male ☐ Female

Email

Salary

Job Position

Select one ▼

Save

SIMPLE SPRING FORM

Staff Id

Full Name

Gender

☒ Male ☐ Female

Email

Salary

Job Position

General Director ▼

Save





CONTROLLER AND MODEL

```
@Controller
@RequestMapping("staff")
public class StaffController {
    @RequestMapping("form")
    public String index(Model model) {
        Staff staff = new Staff();
        model.addAttribute("staff", staff);
        // view - layout
        model.addAttribute("view", "staff/form");
        return "bootstrap";
    }

    @RequestMapping("save")
    public String save(Model model,
        @ModelAttribute("staff") Staff staff) {
        // view - layout
        model.addAttribute("view", "staff/form");
        return "bootstrap";
    }
}
```

```
public class Staff {
    Integer id;
    String fullname;
    Boolean gender;
    String email;
    Double salary;
    String position;

    getters/setters
}
```



SPRING FORM

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<form:form action="staff/save" modelAttribute="staff">
  <form:input path="id" cssClass="form-control"/>
  <form:input path="fullname" cssClass="form-control"/>
  <div class="form-control">
    <form:radio button path="gender" value="true" label="Male"/>
    <form:radio button path="gender" value="false" label="Female"/>
  </div>
  <form:input path="email" cssClass="form-control"/>
  <form:input path="salary" cssClass="form-control"/>
  <form:select path="position" cssClass="form-control">
    <form:option value="" label="Select one"/>
    <form:option value="CEO" label="General Director"/>
    <form:option value="DIR" label="Director"/>
    <form:option value="MAN" label="Manager"/>
    <form:option value="EMP" label="Employee"/>
  </form:select>
  <button class="btn btn-default">Save</button>
</form:form>
```

● **<form:form @modelAttribute>**

● **<form:xyz @path>**



- Buộc dữ liệu từ bean vào form

- ✱ <form:form **modelAttribute**="staff">

- ✱ <form:xyz **path**="fullname">

- Buộc dữ liệu từ form vào bean

- ✱ save(**Staff staff**)

- Buộc dữ liệu 2 chiều

- ✱ save(**@ModelAttribute("staff")** Staff staff)

Spring Form và buộc dữ liệu

ĐỀ MÔ



SPRING FORM TAGS

- | | |
|---|--|
| ● <code><form:form></code> | ➤ <code><form></code> |
| ● <code><form:input/></code> | ➤ <code><input type="text"/></code> |
| ● <code><form:textarea/></code> | ➤ <code><textarea/></code> |
| ● <code><form:checkbox/></code> | ➤ <code><input type='checkbox' /></code> |
| ● <code><form:radiobutton/></code> | ➤ <code><input type='radio' /></code> |
| ● <code><form:hidden/></code> | ➤ <code><input type='hidden' /></code> |
| ● <code><form:password/></code> | ➤ <code><input type='password' /></code> |
| ● <code><form:button/></code> | ➤ <code><button/></code> |
| ● <code><form:select/></code> | ➤ <code><select/></code> |
| ● <code><form:radiobuttons/></code> | ➤ Nhóm radio |
| ● <code><form:checkboxes/></code> | ➤ Nhóm checkbox |



CASE STUDY 2

Nationality

Select one ▼

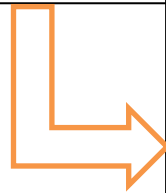
Job Position

☐ Accountant ☐ Employee ☐ Manager ☐ General Director ☐ Directory

Hobbies

☐ Reading ☐ Travelling ☐ Music ☐ Others

Save



Nationality

United States ▼

Job Position

☐ Accountant ☐ Employee ☐ Manager ☒ General Director ☐ Directory

Hobbies

☒ Reading ☐ Travelling ☒ Music ☐ Others

Save



CONTROLLER AND MODEL

```
@Controller
@RequestMapping("staff2")
public class Staff2Controller {
    @RequestMapping("form")
    public String index(Model model) {
        Staff2 staff = new Staff2();
        model.addAttribute("staff", staff);
        // view - layout
        model.addAttribute("view", "staff/form2");
        return "bootstrap";
    }

    @RequestMapping("save")
    public String save(Model model,
        @ModelAttribute("staff") Staff2 staff) {
        // view - layout
        model.addAttribute("view", "staff/form2");
        return "bootstrap";
    }
}
```

```
public class Staff2 {
    Country country;
    String position;
    List<String> hobbies;

    getters/setters
}
```



DATA SOURCES FOR LIST CONTROLS

```
@ModelAttribute("positions")
public Map<String, String> getPositions() {
    Map<String, String> map = new HashMap<>();
    map.put("CEO", "General Director");
    map.put("DIR", "Directory");
    map.put("MAN", "Manager");
    map.put("ACC", "Accountant");
    map.put("EMP", "Employee");
    return map;
}
```

```
@ModelAttribute("hobbies")
public List<String> getHobbies() {
    List<String> list = new ArrayList<>();
    list.add("Reading");
    list.add("Travelling");
    list.add("Music");
    list.add("Others");
    return list;
}
```

```
@ModelAttribute("countries")
public List<Country> getCountries() {
    List<Country> list = new ArrayList<>();
    list.add(new Country("VN", "Việt nam"));
    list.add(new Country("US", "United States"));
    list.add(new Country("UK", "United Kindom"));
    return list;
}
```



SPRING FORM (FILL TO LIST CONTROLS)

```
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
```

```
<form:form action="staff2/save" modelAttribute="staff">
  <form:select path="country.id" class="form-control">
    <form:option value="" label="Select one"/>
    <form:options items="${countries}" itemValue="id" itemLabel="name"/>
  </form:select>
  <div class="form-control">
    <form:radiobuttons path="position" items="${positions}" delimiter=" "/>
  </div>
  <div class="form-control">
    <form:checkboxes path="hobbies" items="${hobbies}" delimiter=" "/>
  </div>
  <button class="btn btn-default">Save</button>
</form:form>
```

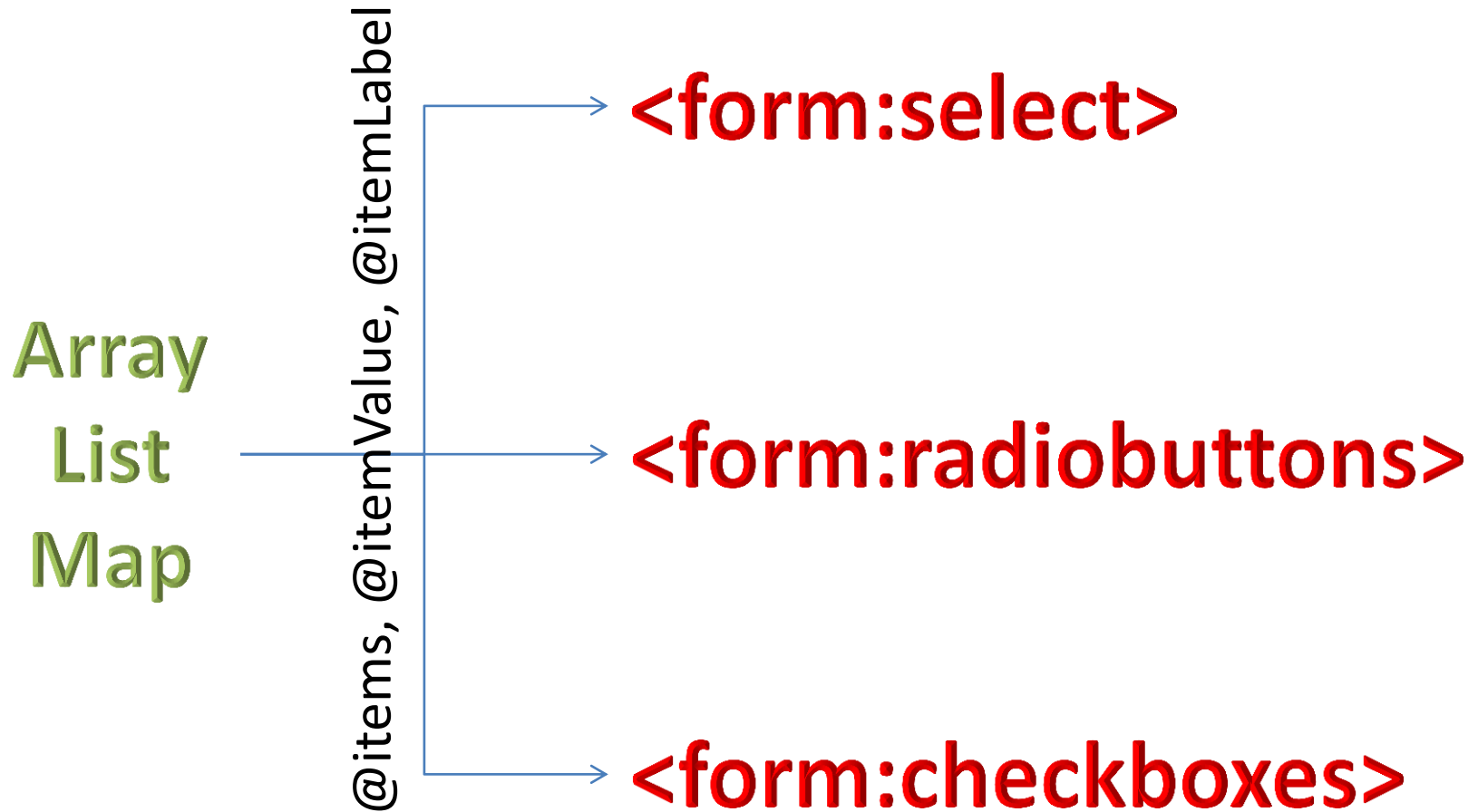
- @items, @itemValue, @itemLabel
- @delimiter

Đổ dữ liệu vào List Control

ĐỀ MÔ



FILL TO LIST CONTROLS





UPLOAD FILE

SEND MAIL WITH ATTACHMENT

From

To

Subject

Content

Attachment

Send

```
<form action="mailer/send"
method="post"
enctype="multipart/form-data">
  <input name="from">
  <input name="to">
  <input name="subject" >
  <textarea name="body" rows="4">
  </textarea>
  <input name="attach" type="file">
  <button>Send</button>
</form>
```



● Cấu hình

spring.servlet.multipart.**max-file-size**=20MB

spring.servlet.multipart.**max-request-size**=20MB

● Controller

```
public String send(  
    @RequestParam("attach") MultipartFile attach) {  
    String filename = attach.getOriginalFilename();  
    File file = new File(app.getRealPath("/docs/" + filename));  
    attach.transferTo(file);  
}
```

PART 2: VALIDATION



CASE STUDY 3

SPRING FORM VALIDATION

Staff Id

Vui lòng nhập mã nhân viên

Full Name

must not be blank

Gender

☐ Male ☐ Female

Vui lòng chọn giới tính

Email

Không đúng định dạng email

Salary

Lương phải lớn hơn 9.5

Job Position

Vui lòng chọn vị trí chức danh

Hobbies

☐ Reading ☐ Travelling ☐ Music ☐ Others

Vui lòng chọn ít nhất một sở thích

Validate



MODEL

```
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



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**: 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 cho một property

- ✱ `<form:errors path="fullname"/>`

- Hiển thị thông báo lỗi cho tất cả

- ✱ `<form:errors path="*/>`

- Cú pháp đầy đủ

- ✱ `<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

Kiểm lỗi Spring Form

ĐỀ MÔ



VALIDATION ANNOTATION

- `@DecimalMax("5"), @DecimalMin("6.5")`
 - ✱ Số nguyên, số thực
- `@Max(5), @Min(6), @Size(min=0, max=1)`
 - ✱ Số nguyên
- `@Digits(integer=3, fraction=2)`
 - ✱ Số ký tự phần nguyên và thập phân
- `@Negative(), @NegativeOrZero(), @Positive(), @PositiveOrZero()`
 - ✱ Số âm, số dương



VALIDATION ANNOTATION

- **@Email()**
 - ✱ Định dạng email
- **@Pattern(regexp="")**
 - ✱ So khớp biểu thức chính qui
- **@NotBlank()**
 - ✱ Chuỗi trống
- **@NotEmpty()**
 - ✱ Chuỗi trống, Collection, Map rỗng
- **@NotNull(), @Null()**
 - ✱ Giá trị null



VALIDATION ANNOTATION

- `@AssertFalse()`, `@AssertTrue()`
 - ✱ Giá trị boolean
- `@Future()`, `@FutureOrPresent()`, `@Past()`, `@PastOrPresent()`
 - ✱ Thời gian quá khứ và tương lai

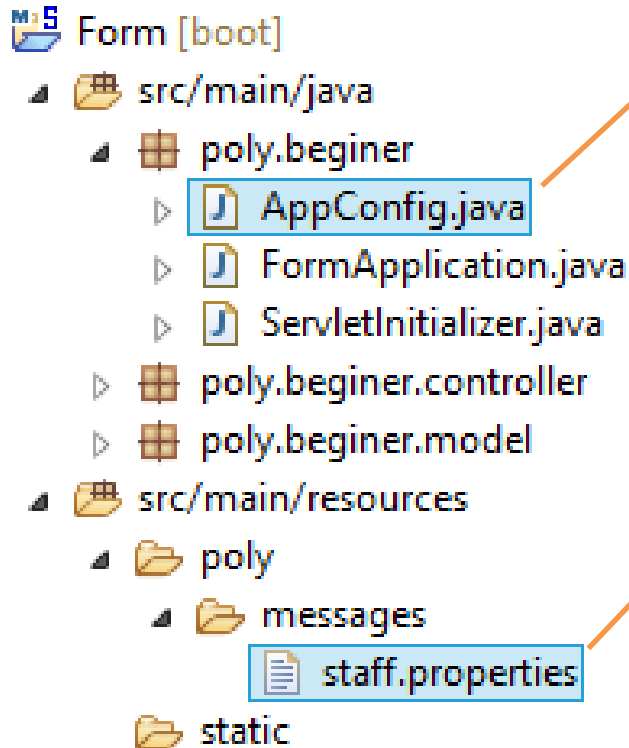


CUSTOMIZING ERROR MESSAGES

- Annotation kiểm lỗi đã định nghĩa sẵn thông báo lỗi mặc định.
- Sử dụng thuộc tính message của annotation để định nghĩa lại thông báo lỗi
 - ✱ @Email(message="Chưa đúng định dạng email")
- Sử dụng tài nguyên (file properties) để định nghĩa thông báo lỗi, rất thuận lợi cho đa ngôn ngữ.



ERROR MESSAGE RESOURCE ORGANIZATION



Nạp tài nguyên
thông báo lỗi vào
ứng dụng

Khao báo thông
báo lỗi



DECLARING ERROR MESSAGES

- Mỗi thông báo lỗi được khai dạng key=value

- ✱ Key: Annotation.bean.field

- ✱ Value: Thông báo lỗi

```
#  
# Thông báo lỗi cho bean staff  
#  
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  
NotNull.staff.salary=Vui lòng nhập lương  
DecimalMin.staff.salary=Lương phải lớn hơn 9.5  
NotEmpty.staff.position=Vui lòng chọn vị trí chức danh  
NotEmpty.staff.hobbies=Vui lòng chọn ít nhất một sở thích  
typeMismatch=Sai kiểu dữ liệu
```



LOADING ERROR MESSAGES

- ReloadableResourceBundleMessageSource giúp nạp các file tài nguyên vào ứng dụng
- setBasenames(...) chỉ ra danh sách các file tài nguyên cần nạp (*không bao gồm .properties*)

```
@Configuration
public class AppConfig implements WebMvcConfigurer{
    @Bean(name="messageSource")
    public MessageSource getMessageSource() {
        ReloadableResourceBundleMessageSource msg
            = new ReloadableResourceBundleMessageSource();
        msg.setDefaultEncoding("UTF-8");
        msg.setBasenames("classpath:poly/messages/staff");
        return msg;
    }
}
```


Tùy biến thông báo lỗi

ĐỀ MÔ



SUMMARY

- Databinding
- Spring Form
- Up/Down/Two way Databinding
- Fill data to list Controls
- Validation
 - ✱ Bean: Annotations
 - ✱ Controller: @Valid và BindingResult
 - ✱ View: <form:errors>
 - ✱ Messages: messageSource configuration

