

Bài 4

Data Binding Form

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT

Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài "Views Thymleaf"

Tóm tắt lại các phần đã học từ bài "Views Thymeleaf"

- Thao tác *được* với form trong ứng dụng Spring MVC
- Trình bày *được* cơ chế Data Binding
- *Sử dụng được* các thẻ thành phần của form

Thảo luận

Form và Data Binding

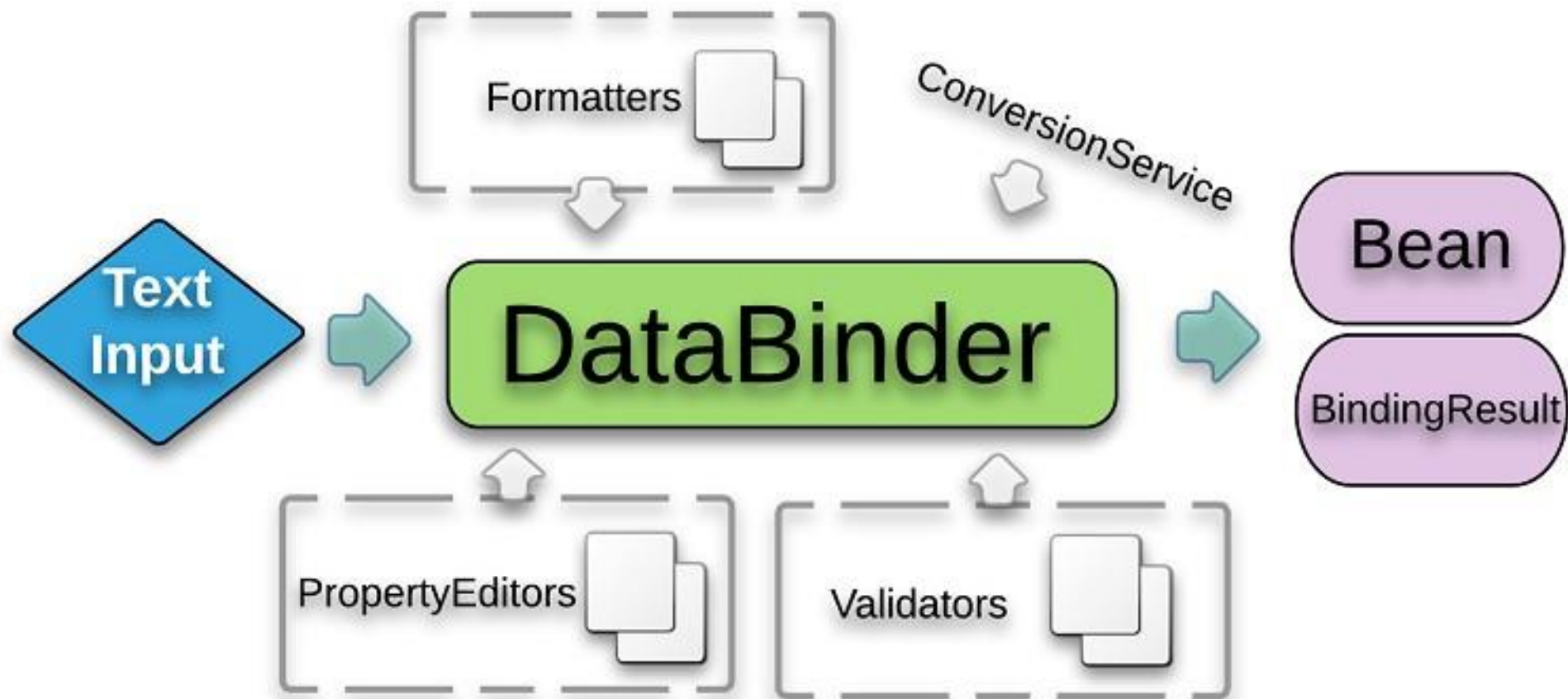
- Form là cơ chế để cho phép người dùng nhập dữ liệu
- Spring MVC cung cấp các thẻ trong thư viện *spring-form.tld* để thao tác với form
- Khai báo *spring-form.tld*:

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

- *spring-form.tld* cung cấp thẻ form và các thẻ thành phần khác để thao tác với form
- *spring-form.tld* cung cấp các cơ chế để kiểm tra lỗi, tùy biến giao diện và triển khai quốc tế hoá (i18n)

- Data Binding là cơ chế liên kết dữ liệu đầu vào (hoặc đầu ra) với các đối tượng model
- Data Binding giúp cho việc tương tác với dữ liệu trở nên dễ dàng
- Sử dụng Data Binding, các form đều được liên kết với một đối tượng biểu diễn dữ liệu ở phía sau
- Khi tương tác với form, dữ liệu trên form sẽ được tự động chuyển đổi thành các thuộc tính của đối tượng liên kết với nó
- Data Binding hỗ trợ chuyển đổi dữ liệu (data conversion) và validate dữ liệu
- Data Binding trong Spring hoạt động dựa trên Data Binder

Cơ chế hoạt động của Data Binding



Thẻ form

- Thẻ form được sử dụng để tạo một form
- Ngoài các thuộc tính cơ bản của một thẻ form trong HTML, thẻ form còn có thêm thuộc tính quan trọng:
 - *commandName* hoặc *modelAttribute*: Tên của model được liên kết với form
- Thuộc tính *commandName* và *modelAttribute* có tác dụng giống nhau (*commandName* là cách dùng cũ, *modelAttribute* là cách dùng mới)
- Ví dụ:


```
<form:form modelAttribute="user" action="/create-user" method="post">
</form:form>
```


Các thẻ của spring-form.tld

- spring-form.tld hỗ trợ các thẻ: button, checkbox, checkboxes, errors, form, hidden, input, label, option, options, password, radiobutton, radiobuttons, select, textarea
- Một số thuộc tính quan trọng:
 - *path*: tên của thuộc tính được liên kết với trường hiện tại
 - *items*: danh sách các hạng mục của các trường như select, options, radiobuttons, checkboxes

Các thẻ input, password, textarea, hidden

- Thuộc tính quan trọng
 - path: tên thuộc tính của model được liên kết với trường input

- Ví dụ:

```
<form:form modelAttribute="user" action="/create-user" method="post">  
  <form:hidden path="id"/>  
  <form:input path="name"/><br/>  
  <button type="submit">Create  
  user</button>  
</form:form>
```

```
public class User {  
    private int id;  
    private String  
    name;  
    private String gender;  
    private List<String>  
    hobbies;
```

```
//Constructors, Getters & Setters  
}
```

The checkbox và checkboxes

- Ví dụ sử dụng checkbox và checkboxes:

```
<form:form modelAttribute="user" action="/create-user"
  method="post"> Name: <form:input path="name"/><br/>
  Is admin: <form:checkbox path="admin"/><br/>
  Hobbies: <form:checkboxes path="hobbies" items="${hobbiesArray}"/><br/>
  <button type="submit">Create user</button>
</form:form>
```

```
public class User {

    private int id;
    private String
    name;
    private String gender;
    private boolean admin;
    private List<String>
    hobbies;
    //Constructors, Getters & Setters
}
```

Thẻ radiobutton và radiobuttons

- Ví dụ sử dụng thẻ radiobuttons:

```
<form:form modelAttribute="user" action="/create-user" method="post">
```

```
...
```

```
    Type: <form:radiobuttons path="type" items="${typesArray}"/><br/>
```

```
...
```

```
</form:form>
```

The select, option, options

- Ví dụ sử dụng thẻ select:

```
<form:form modelAttribute="user" action="/create-user" method="post">
```

```
...
```

```
    Type: <form:select path="type" items="${typesArray}"/><br/>
```

```
...
```

```
</form:form>
```

Demo

Form và Data Binding

Thảo luận

Upload file

Form hỗ trợ upload file

- Form hỗ trợ upload file thông qua input có type là file
- Form cần có thêm thuộc tính *enctype* với giá trị là *multipart/form-data*

```
<form:form modelAttribute="userForm" action="/create-user"  
method="post" enctype="multipart/form-data">  
    <form:input type="file" path="avatar"/>  
</form:form>
```


- Lớp `org.springframework.web.multipart.MultipartFile` đại diện cho một file được upload lên
- Một số phương thức quan trọng:
 - `getOriginalFilename()`: Trả về tên của file
 - `getInputStream()`: Trả về Input Stream để thao tác với file
 - `getSize()`: Trả về kích thước của file
 - `transferTo(File)`: Chuyển file vừa upload đến một vị trí nhất định

- Spring cung cấp lớp MultipartResolver để làm việc với multipart form
- MultipartResolver không trực tiếp xử lý multipart form, mà có thể dựa vào 1 trong 2 cách sau:
 - Sử dụng thư viện Commons FileUpload
 - Sử dụng cơ chế xử lý multipart của Servlet 3.0

Cấu hình sử dụng Commons FileUpload

- Cấu hình dependency:

```
dependencies {  
    compile group: 'commons-fileupload', name: 'commons-fileupload', version: '1.3.3'  
}
```

- Cấu hình bean multipartResolver:

@Bean

```
public CommonsMultipartResolver multipartResolver(){  
    CommonsMultipartResolver multipartResolver = new  
    CommonsMultipartResolver();  
    multipartResolver.setMaxUploadSizePerFile(10000000);  
    return multipartResolver;  
}
```

Cấu hình sử dụng Servlet 3.0 (1)

- Cấu hình dependency:

```
dependencies {  
    compile group: 'javax.servlet', name: 'javax.servlet-api', version: '3.1.0'  
}
```

- Cấu hình bean multipartResolver

```
@Bean  
public StandardServletMultipartResolver multipartResolver(){  
    return new StandardServletMultipartResolver();  
}
```

Cấu hình sử dụng Servlet 3.0 (2)

- Cấu hình Multipart:

```
public class AppInitializer extends
    AbstractAnnotationConfigDispatcherServletInitializer { @Override
    protected void customizeRegistration(ServletRegistration.Dynamic registration) {
        MultipartConfigElement multipartConfigElement = new
        MultipartConfigElement(FileUtils.UPLOAD_LOCATION);
        registration.setMultipartConfig(multipartConfigElement);
    }
}
```

Demo

Upload file

- Data Binding là cơ chế liên kết dữ liệu từ giao diện đến đối tượng ở phía sau
- Spring cung cấp spring-form.tld với các thẻ form thông dụng
- Có thể upload file sử dụng Commons FileUpload hoặc Servlet 3.0

Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: *Database ORM*