



BUILDING ACTION METHOD

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



MỤC TIÊU

- Ánh xạ được url với method
- Phân biệt được các kết quả trả về của method
- Sử dụng được các hình thức chia sẻ qua Model và RedirectAttributes
- Tiếp nhận và xử lý dữ liệu từ người dùng





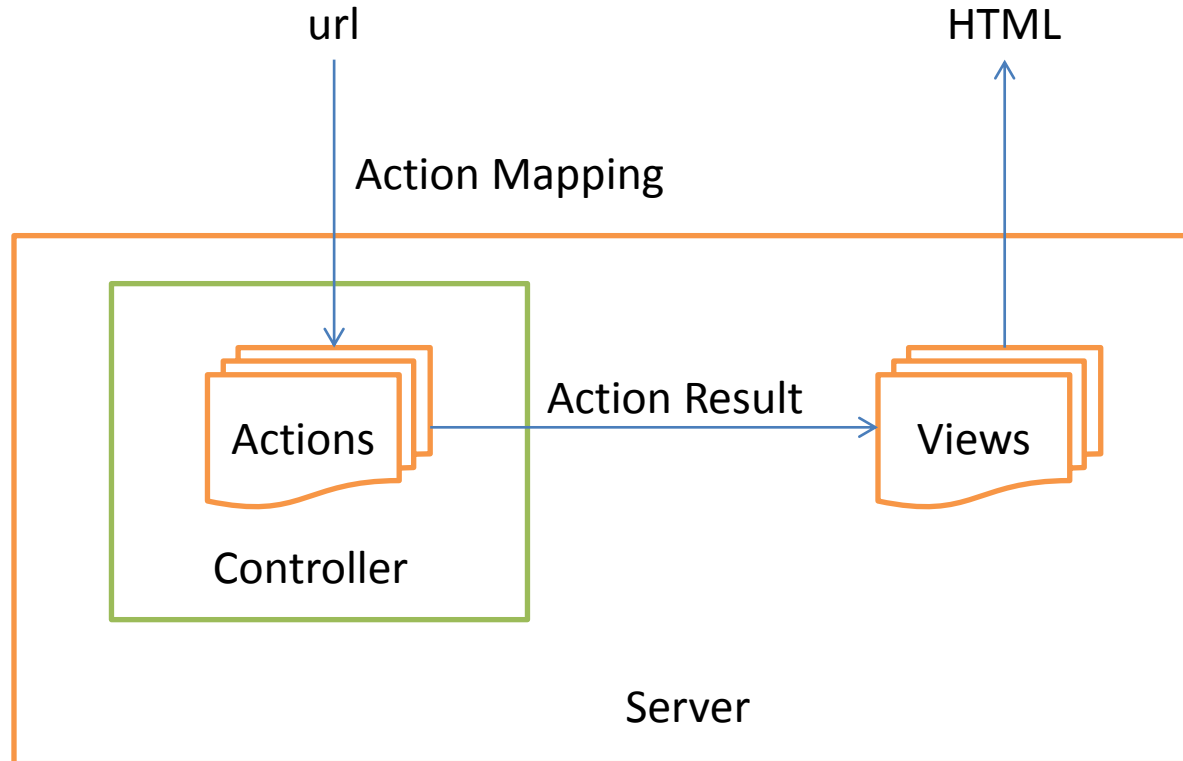
NỘI DUNG

- Các annotation ánh xạ url-method
- Ánh xạ phân biệt POST và GET
- Ánh xạ phân biệt tham số
- Sử dụng @ModelAttribute để chia sẻ dữ liệu qua Model
- Kết quả trả về của action method
- Truyền dữ liệu redirect
- Tiếp nhận và xử lý dữ liệu từ người dùng





ACTION METHOD INTRODUCTION



- Action input?
- Action output?
- Passing data to View?



ACTION MAPPING

- Spring MVC cung cấp 3 annotation để thực hiện ánh xạ URL vào Action Method
 - ✱ **@RequestMapping**(value, params)
 - Không phân biệt POST hay GET
 - ✱ **@GetMapping**(value, params)
 - Chỉ với GET
 - ✱ **@PostMapping**(value, params)
 - Chỉ với POST



@RequestMapping

```
@Controller
@RequestMapping("home")
public class HomeController {
    @RequestMapping("index")
    public String index() {
        return "index";
    }

    @RequestMapping("about")
    public String about() {
        return "about";
    }
}
```



```
@Controller
public class HomeController {
    @RequestMapping("home/index")
    public String index() {
        return "index";
    }

    @RequestMapping("home/about")
    public String about() {
        return "about";
    }
}
```



@PostMapping và @GetMapping

● GET

- ✱ Hiển thị form

● POST

- ✱ Xử lý dữ liệu

```
@Controller
@RequestMapping("account")
public class AccountController {
    @GetMapping("login")
    public String login() {
        return "login";
    }
    @PostMapping("login")
    public String login(MyBean bean) {
        return "login";
    }
}
```



MAPPING WITH PARAMETERS

```
@Controller
@RequestMapping("url")
public class MyController {
    @RequestMapping()
    public String method1() {
        return "index";
    }

    @RequestMapping(params="btnInsert")
    public String method2() {
        return "index";
    }

    @RequestMapping(params="btnUpdate")
    public String method3() {
        return "index";
    }
}
```

- Ánh xạ 1 URL cho nhiều method
- Phân biệt method bằng cách sử dụng param

CÁC KỸ THUẬT ÁNH XẠ URL-METHOD

ĐỀ MÔ



MODEL

- Model là nơi chứa dữ liệu cung cấp cho view
- Có 3 cách đặt dữ liệu vào Model
 - ✱ `Model.addAttribute(name, value)`
 - ✱ `@ModelAttribute(name)`
 - Đặt trước đối số của Action Method
 - Đặt trên phương thức có trả về



MODEL

```
@RequestMapping("url")
public String method(Model model) {
    model.addAttribute("user", new User());
    return "index";
}

@RequestMapping("url")
public String method(@ModelAttribute("user") User user) {
    return "index";
}

@ModelAttribute("user")
public User getUser() {
    return new User();
}
```

```
public class User {
    private String id;
    private String password;
    public String getId() {...}
    public void setId(String id) {...}
    public String getPassword() {...}
    public void setPassword(String password) {...}
}
```

3 CÁCH CHIA SẺ DỮ LIỆU QUA MODEL

ĐỀ MÔ

ACTION METHOD RESULT

```
@RequestMapping("url")  
public String method() {  
    return "view";  
}
```

Tên view

```
@RequestMapping("url")  
public void method() {  
}
```

Tên view chính là url

```
@ResponseBody  
@RequestMapping("url")  
public String method() {  
    return "Hello Spring MVC";  
}
```

Dữ liệu trực tiếp, không
qua ViewResolver

```
@RequestMapping("url")  
public String method() {  
    return "forward:/other-url";  
}
```

Chuyển tiếp sang url
khác

```
@RequestMapping("url")  
public String method() {  
    return "redirect:/other-url";  
}
```

Chuyển hướng sang url
khác

ACTION RESULTS

ĐỀ MÔ



USER DATA HANDLING

- Nhận tham số
 - ✱ HttpServletRequest
 - ✱ @RequestParam
 - ✱ JavaBean
- Dữ liệu từ đường dẫn
 - ✱ @PathVariable
- Dữ liệu từ cookie
 - ✱ @CookieValue



USING HTTPServletRequest

```
@Autowired
HttpServletRequest request;

@RequestMapping("url")
public String method() {
    String id = request.getParameter("id");
    String password = request.getParameter("password");
    return "index";
}
```

- Nhược điểm: Phải tự chuyển kiểu



USING @REQUESTPARAM

- @RequestParam(name, defaultValue)

- ✱ Name: tên tham số

- ✱ defaultValue: giá trị mặc định

- Ví dụ

```
@RequestMapping("url")  
public String method(@RequestParam(name="id", defaultValue="123") String id,  
                     @RequestParam("password") String password) {  
    return "index";  
}
```



USING JAVA BEAN

```
@RequestMapping("url")
public String method(User user) {
    return "index";
}
```

```
public class User {
    private String id;
    private String password;
    public String getId() {...}
    public void setId(String id) {...}
    public String getPassword() {...}
    public void setPassword(String password) {...}
}
```

- Đơn giản, tự chuyển kiểu, tự kiểm lỗi

PARAMETER

ĐỀ MÔ



USING @PATHVARIABLE

```
@RequestMapping("url/{id}/{password}")  
public String method(@PathVariable(name="id") String id,  
                     @PathVariable("password") String password ) {  
    return "index";  
}
```

- url/ngghiemn/iloveyou
- Phù hợp để SEO



USING @COOKIEVALUE

```
@Controller
public class MyController{
    @GetMapping("url")
    public String method(
        @CookieValue(value="uid", defaultValue="") String id)
    {...}
    ...
}
```

PART AND COOKIE

ĐỀ MÔ



- Mapping

- ✱ @RequestMapping, @GetMapping, @PostMapping

- Data Sharing

- ✱ Model, @ModelAttribute

- Action Result

- ✱ @ResponseBody, redirect and forward

- User Data

- ✱ @RequestParam, JavaBean, @PathVariable, @CookieValue

