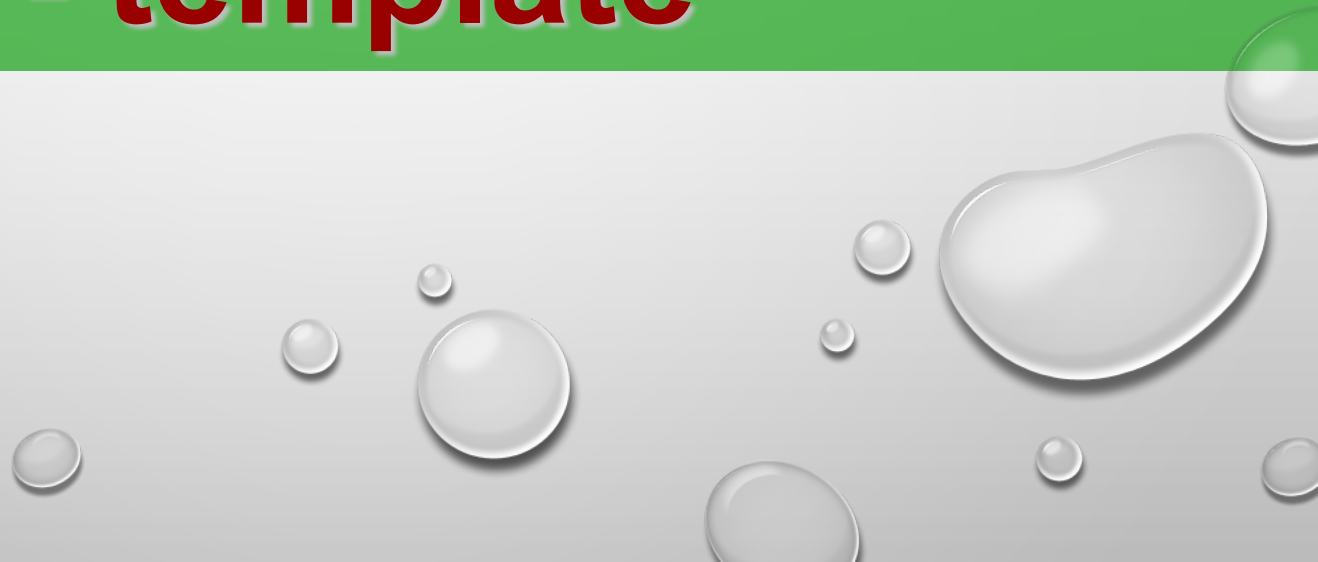


CHƯƠNG 18 –SPRING

- template



Nội dung

1. Khai báo trong Spring Framework
2. Sử dụng template trong Spring Framework với Apache Tiles
3. Ví dụ

1. Khai báo Annotation

Giới thiệu

- Làm việc với Spring Framework thì việc sử dụng các Annotation để khai báo trong Spring Framework là chuyện bạn thường phải thực hiện

Khai báo Annotation

- **@Controller**: Trong Spring Framework thì @Controller được dùng để đánh dấu đây là lớp xử lý, lớp điều khiển (controller) như ví dụ bên dưới đây
- **@RequestMapping**: để nhằm khai báo nhằm gọi đến địa chỉ URL tương ứng để xử lý, cụ thể bạn hãy xem ví dụ ngay bên dưới

Khai báo Annotation

```
12
13 @Controller
14 @RequestMapping (value="/hello")
15 public class HelloController {
16     //http://localhost:8080/spingmvcintro/hello
17     @RequestMapping (method = RequestMethod.GET)
18     public String index(ModelMap m){
19         m.addAttribute("name", "PhamvanHao");
20         return "hello";
21     }
22
23
24     //http://localhost:8080/spingmvcintro/hello
25     @RequestMapping (value = "/xinchao" , method = RequestMethod.GET)
26     public String xinchao(ModelMap m){
27         m.addAttribute("name", "PhamvanHao");
28         m.put("name2", "Pham van Hao");
29         return "xinchao";
30     }
31 }
```

Khai báo Annotation

- **@PathVariable**: lấy tham số thông qua URL thì với chú thích @PathVariable sẽ hiệu quả hơn nhiều, vậy sử dụng chú thích @PathVariable như thế nào

```
// http://localhost:8080/spingmvcintro/hello/show/day la noi dung
@RequestMapping(value = "/show/{content}", method = RequestMethod.GET)
public String hienthi(@PathVariable(value = "content") String content, ModelMap mm) {
    mm.put("nd", content);
    return "show";
}
```

Khai báo Annotation

- **@ModelAttribute**: Chú thích @ModelAttribute trong Spring Framework sẽ hữu ích đối với trường hợp lấy dữ liệu với nhiều tham số. Chẳng hạn muốn lấy dữ liệu từ form với các tham số mã sản phẩm, tên sản phẩm, giá bán, mô tả... thì sử dụng @ModelAttribute sẽ tốt hơn nếu như muốn sử dụng @RequestParam
- @RequestParam:

```
@RequestMapping(value = "/luu-du-lieu.html", method = RequestMethod.POST)
public String find(@RequestParam String maSanPham, @RequestParam String tenSanPham,
    @RequestParam double giaBan, @RequestParam String tuKhoa, @RequestParam String moTaNgan,
    @RequestParam String moTaChiTiet, @RequestParam String tinhTrang) {
    // xử lý
    return "ket_qua";
}
```

- @ModelAttribute:

```
@RequestMapping(value = "/luu-du-lieu.html", method = RequestMethod.POST)
public String find(@ModelAttribute SanPham sanPham) {
    // xử lý
    return "ket_qua";
}
```

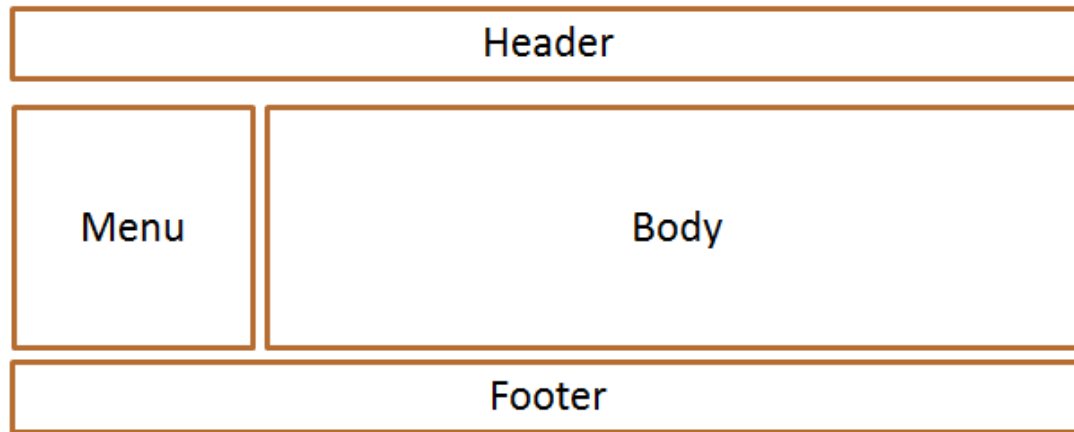

Khai báo Annotation

- **@Autowired**: Tự động liên kết các Bean lại với nhau
- **@Repository**: dùng để đánh dấu lớp DAO
- **@Service**: dùng để đánh dấu lớp Service
- **@Transactional**: được dùng để đánh dấu các lớp sử dụng đến transaction
- **@Scope**: mặc định trong Spring, để thay đổi phạm vi ta sử dụng @scope
- **@Component**: dùng để scan components
- ...

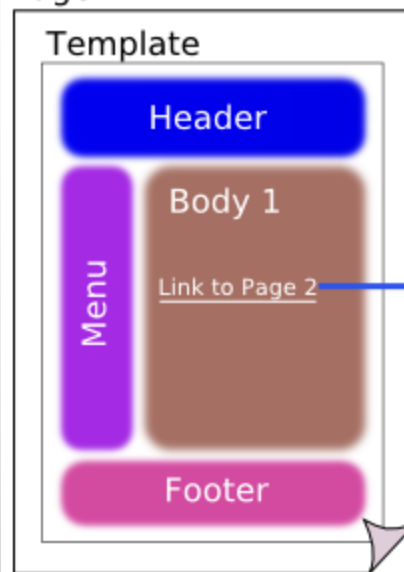
2. Sử dụng template trong Spring Framework với Apache Tiles

Các thành phần

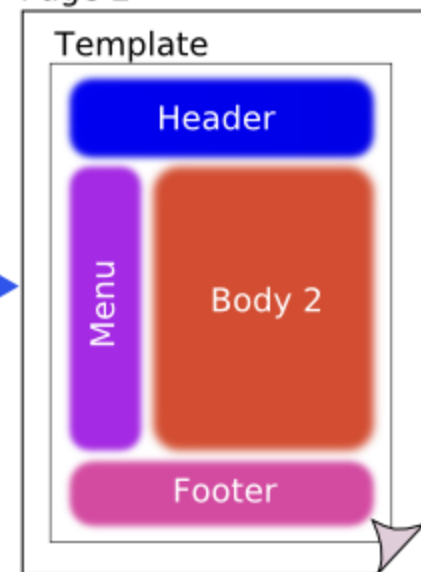
- Để đơn giản hóa việc phát triển giao diện người dùng của ứng dụng web thiết kế theo template framework, có thể sử dụng thư viện Apache Tiles.
- Một trang web được coi là một mái nhà, nó được ghép lại từ các viên ngói, một viên ngói ở đây là một file jsp (jsp fragment).
Apache Tiles giúp bạn định nghĩa ra một khuôn mẫu (Template) để ghép các thành phần (các mảnh jsp) lại với nhau để thành một trang web.



Page 1



Page 2



Lợi ích khi sử dụng template với Apache Tiles

- **Khả năng tái sử dụng:** Chúng ta có thể tái sử dụng một thành phần trong nhiều trang như header, footer, menu...
- **Dễ dàng thay đổi bố cục:** Nhờ sử dụng một trang mẫu (template) mà bạn có thể thay đổi bố cục giao diện website một cách đơn giản, thuận tiện và không phải viết lại code quá nhiều.

Triển khai

- **Bước 1:**

- Tạo một ứng dụng web bằng Spring Framework.
Sau khi tạo xong bạn cần tải bộ thư viện Apache Tiles và giải nén import vào ứng dụng (Có thể sử dụng Maven).

- **Bước 2:**

- Tạo các tập tin cấu hình **templates.xml (tiles.xml)** như là một khuôn mẫu (template) chung cho các màn hình giao diện.
- Tạo các file View: Giao diện mẫu sẽ có phần header, menu, footer và phần content có thể thay đổi.

- **Bước 3:**

- Build và chạy ứng dụng

Ví dụ demo

HẾT CHƯƠNG 18