[Bài đọc] Views và View Resolvers

**Views và View Resolvers trong Spring MVC**

Có hai interface trong Spring MVC phù hợp với core của hệ thống templating:

org.springframework.web.servlet.View  
org.springframework.web.servlet.ViewResolver

Đây là các view model trong các ứng dụng. Nó cho phép sửa đổi, xác định trước hành vi bằng cách định nghĩa nó là các bean. Các view phụ trách việc hiển thị giao diện HTML, thường là thực thi một số template engine như Thymeleaf.

Các ViewResolver (trình phân giải view) là các đối tượng chịu trách nhiệm lấy các đối tượng View cho một hành động và vị trí cụ thể. Thông thường, các controller yêu cầu ViewResolvers chuyển tiếp đến một view với một tên cụ thể (một chuỗi được trả về bởi phương thức). Sau đó, tất cả các view resolver trong ứng dụng thực thi theo thứ tự cho đến khi một trong số đó có thể phân giải view trong trường hợp một đối tượng View được trả về và điều khiển được truyền tới nó cho việc render HTML.

Lưu ý rằng, không phải tất cả các trang trong ứng dụng của chúng ta đều được định nghĩa là Views, chỉ những hành vi mà chúng ta muốn phi chuẩn hoặc được cấu hình theo một cách cụ thể (ví dụ, bằng cách gắn một số bean đặc biệt với nó). Nếu một ViewResolver yêu cầu một view mà không có bean tương ứng, một đối tượng View mới được tạo ra một cách đặc biệt và được trả về.

Một cấu hình điển hình cho JSP + ViewResolver JSPL trong ứng dụng Spring MVC trước đây trông như sau:

<**bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 <**property name="viewClass" value="org.springframework.web.servlet.view.JstlView"** />  
 <**property name="prefix" value="/WEB-INF/jsps/"** />  
 <**property name="suffix" value=".jsp"** />  
 <**property name="order" value="2"** />  
 <**property name="viewNames" value="\*jsp"** />  
</**bean**>

Cách mà JSP + ViewResolver JSPL trong ứng dụng Spring MVC được cấu hình:

* viewClass thiết lập lớp các thể hiện của View. Điều này là cần thiết cho một trình phân giải JSP, nhưng nó sẽ không cần thiết khi chúng ta đang làm việc với Thymeleaf.
* prefix và suffix hoạt động theo cách tương tự với các thuộc tính cùng tên trong các đối tượng TemplateResolver của Thymeleaf.
* order thiết lập thứ tự mà ViewResolver sẽ được truy vấn trong chuỗi.
* viewNames cho phép định nghĩa các tên view sẽ được phân giải bởi ViewResolver này.

**Views và View Resolvers trong Thymeleaf**

Thymeleaf cung cấp các triển khai cho hai interface được đề cập ở trên:

org.thymeleaf.spring4.view.ThymeleafView  
  
org.thymeleaf.spring4.view.ThymeleafViewResolver

Hai lớp này sẽ chịu trách nhiệm xử lý các template Thymeleaf như là kết quả của việc thực thi các controller.

Cấu hình của Thymeleaf View Resolver rất giống với cấu hình của JSP:

@Bean  
public ThymeleafViewResolver viewResolver(){  
ThymeleafViewResolver viewResolver = new ThymeleafViewResolver();  
viewResolver.setTemplateEngine(templateEngine());  
// NOTE 'order' and 'viewNames' are optional  
viewResolver.setOrder(1);  
viewResolver.setViewNames(new String[] {".html", ".xhtml"});  
return viewResolver;  
}

Hoặc trong XML:

<**bean class="org.thymeleaf.spring4.view.ThymeleafViewResolver"**>  
 <**property name="templateEngine" ref="templateEngine"** />  
 *<!-- NOTE 'order' and 'viewNames' are optional -->* <**property name="order" value="1"** />  
 <**property name="viewNames" value="\*.html,\*.xhtml"** />  
</**bean**>

Lưu ý rằng, chúng ta không cần tham số prefix hoặc suffix, bởi vì chúng đã được chỉ định tại Template Resolver (lần lượt được truyền tới Template Engine).

Và điều gì xảy ra nếu chúng ta muốn định nghĩa một bean “View” và thêm một số biến tĩnh vào nó? Chúng ta chỉ cần xác định một *prototype* (nguyên mẫu) bean cho nó:

@Bean  
@Scope("prototype")  
public ThymeleafView mainView() {  
ThymeleafView view = new ThymeleafView("main"); // templateName = 'main'  
view.setStaticVariables(  
Collections.singletonMap("footer", "The ACME Fruit Company"));  
return view;  
}

Bằng cách này, bạn có thể thực thi bean “View” bằng cách chọn nó bằng tên bean (mainView, trong trường hợp này).

Last modified: Monday, 28 January 2019, 9:35 AM

[Previous Activity](https://james.codegym.vn/mod/page/view.php?id=8910&forceview=1)

Top of Form

Jump to...

Bottom of Form