[Bài đọc] Giới thiệu về Thymeleaf

**1. Template là gì?**

Template là một cách thức thông dụng để hiển thị kết quả của một request về cho người dùng. Template định nghĩa một khuôn mẫu sẵn, sau đó ứng dụng sẽ đổ dữ liệu vào template để tạo ra một response thực tế và trả về cho người dùng.

Template giúp cho việc xây dựng giao diện trở nên dễ dàng hơn thông qua việc tách các mã nguồn nghiệp vụ (chứa trong controller) với mã nguồn giao diện (chứa trong các file template).



Trong hình minh hoạ ở trên, chúng ta có một file template, trong đó có dòng chữ "Hello ${name}". Giá trị **name** ở đây là chưa tồn tại, thay vào đó nó chỉ đánh dấu rằng sau này sẽ được thay thế bởi giá trị của một biến **name** nào đó.

Sau đó, trong controller sẽ có một biến name, giá trị của biến này sẽ được thay thế cho biểu thức ${name} ở trong template, kết quả của việc này là nội dung của response được trả về cho người dùng.

Có nhiều template engine khác nhau, thông dụng nhất có thể kể đến: JSP, Freemaker, Thymeleaf...

**2. Thymeleaf là gì ?**

Thymeleaf là một Java template engine phía máy chủ, nó có thể làm việc với cả hai môi trường, web và non-web, nó có khả năng xử lý HTML, XML, JavaScript, CSS và thậm chí cả văn bản thuần túy.

Mục tiêu chính của Thymeleaf là mang lại các template tự nhiên, đồng nhất, đơn giản (nature templates) và dễ dàng bảo trì. Để đạt được điều này, Thymeleaf xây dựng trên khái niệm về các mầu tự nhiên (natural templates) để chèn logic của nó vào các file template mà không ảnh hưởng đến template. Điều này giúp thu hẹp khoảng cách giữa các nhóm thiết kế và phát triển.

 Ngay từ đầu, Thymeleaf đã được thiết kế với tiêu chuẩn Web - đặc biệt là HTML5 - cho phép bạn tạo các mẫu xác nhận (validating templates) đầy đủ.

**3. Những loại template Thymeleaf có thể xử lý ?**

Thymeleaf cho phép bạn xử lý sáu loại template, mỗi loại được gọi là một chế độ Template, bao gồm:

* HTML
* XML
* TEXT
* JAVASCRIPT
* CSS
* RAW

Có hai chế độ template đánh dấu (markup) là HTML và XML, ba chế độ template văn bản là TEXT, JAVASCRIPT và CSS và một chế độ template trống là RAW.

Chế độ template HTML sẽ cho phép bất kỳ loại đầu vào HTML nào, bao gồm HTML5, HTML 4 và XHTML. Việc kiểm tra tính hợp lệ hoặc kiểm tra tính đúng đắn sẽ được thực hiện và mã / cấu trúc mẫu sẽ được tôn trọng ở mức độ lớn nhất có thể trong đầu ra.

Chế độ template XML sẽ cho phép đầu vào XML. Trong trường hợp này, mã sẽ được định dạng tốt - không có thẻ không rõ ràng, không có thuộc tính không được trích dẫn, v.v. - và trình phân tích cú pháp sẽ ném ra ngoại lệ nếu tìm thấy các vi phạm. Lưu ý rằng, không có xác thực (đối với DTD hoặc Lược đồ XML) sẽ được thực hiện.

Chế độ template TEXT sẽ cho phép sử dụng cú pháp đặc biệt cho các template không đánh dấu (markup). Ví dụ về các template như vậy có thể là email hoặc các tài liệu thông thường. Lưu ý rằng, các template HTML hoặc XML cũng có thể được xử lý dưới dạng TEXT, trong trường hợp này, chúng sẽ không được coi là đánh dấu (markup) và tất cả các thẻ, DOCTYPE, comment, v.v. sẽ được coi là văn bản thuần túy.

Chế độ template JAVASCRIPT sẽ cho phép xử lý các tệp JavaScript trong ứng dụng Thymeleaf. Điều này có nghĩa là có thể sử dụng dữ liệu mô hình bên trong các tệp JavaScript theo cùng cách thực hiện trong các tệp HTML. Chế độ template JAVASCRIPT được coi là văn bản và do đó, nó sử dụng cú pháp đặc biệt giống như chế độ template TEXT.

Chế độ template CSS sẽ cho phép xử lý các tệp CSS liên quan đến ứng dụng Thymeleaf. Tương tự như chế độ template JAVASCRIPT, chế độ template CSS cũng là một chế độ văn bản và sử dụng cú pháp xử lý đặc biệt từ chế độ template TEXT.

Chế độ template RAW sẽ đơn giản là không xử lý các template. Có nghĩa là, nó được sử dụng để chèn các tài nguyên không bị ảnh hưởng (tệp, phản hồi URL, v.v.) vào các template đang được xử lý. Ví dụ, các tài nguyên bên ngoài, không được kiểm soát ở định dạng HTML có thể được đưa vào các template ứng dụng, sẽ là an toàn khi biết rằng bất kỳ mã Thymeleaf nào mà các tài nguyên này có thể include sẽ không được thực hiện.

Last modified: Tuesday, 29 January 2019, 2:28 PM