[Bài đọc] I18N (Internationalization)

**Quốc tế hoá trong Spring MVC**

Để i18n và l10n ứng dụng, yêu cầu phải:

1. Tách các thành phần văn bản thành các tệp properties
2. Làm cho ứng dụng có khả năng chọn và đọc chính xác các trường từ tệp thuộc tính.

**Tách các thành phần văn bản thành các tệp thuộc tính**

Sau đây là phiên bản cho tiếng Anh của một tệp properties. Nó có hai khóa: greetings và farewell:

Greetings=Hello

farewell=Goodbye

Phiên bản tiếng Đức:

greetings=Hallo

farewell=Tschüß

Và tiếng Trung (với mã ký tự tiếng Trung trong bảng mã Unicode, không phải là lỗi):

greetings=\u4f60\u597d

farewell=\u518d\u89c1

Các ngôn ngữ như tiếng Trung thường được soạn thảo bản dịch bằng trình soạn thảo thông thường, sau đó sử dụng công cụ để chuyển nội dung thành Unicode.

Các tệp properties cần có chung một tên cơ sở, gọi là base name, có thể là bất kỳ tên nào, theo sau tên cơ sở là dấu gạch dưới, sau đó là mã ngôn ngữ và nếu cần, sau đó sẽ là một dấu gạch dưới khác và mã quốc gia. Ví dụ, các tệp trong ví dụ trên có thể được đặt tên như sau:

MyResources\_en\_US.properties

MyResources\_de\_DE.properties

MyResources\_zh\_CN.properties

Nhóm các tệp properties có chung tên cơ sở được gọi là một resource bundle, và được đối tượng hóa bằng lớp java.util.ResourceBundle. Lớp này cho phép chọn và đọc tệp properties xác định theo ngôn ngữ, và tra cứu giá trị các trường trong tệp đó. ResourceBundle là một lớp trừu tượng, nhưng có các phương thức static getBundlle trả về một đối tượng cụ thể của một lớp con cụ thể.

**Đọc các tệp properties**

Như trên đã nói, ResourceBundle là một lớp trừu tượng. Tuy nhiên, bạn có thể lấy một đối tượng của ResourceBundle bằng cách gọi phương thức static getBundle của nó. Nhận diện phương thức của nó là:

public static ResourceBundle getBundle(java.lang.String baseName)

public static ResourceBundle getBundle(java.lang.String baseName, Locale locale)

Ví dụ:

ResourceBundle rb = ResourceBundle.getBundle("MyResources", Locale.US);

Câu lệnh trên sẽ tải ResourceBundle bới các giá trị trong tệp properties tương ứng.

Nếu không tìm thấy tệp properties phù hợp, getBundle sẽ sử dụng tệp properties mặc định, là tệp có tên theo dạng <base name>.properties. Ví dụ MyResources.properties. Nếu không tìm thấy tệp mặc định, một ava.util.MissingResourceException sẽ được throw.

Để đọc một giá trị, sử dụng phương thức getString của lớp ResourceBundle:

public java.lang.String getString (java.lang.String key)

Nếu không tìm thấy thông điệp tương ứng với key, một ` java.util.MissingResourceException` sẽ được throw.

Trong Spring MVC, lập trình viên không cần phải làm việc trực tiếp với ResourceBundle. Thay vào đó là sử dụng bean messageSource để báo cho Spring MVC biết nơi lưu trữ các tệp properties. Ví dụ cấu hình dưới đây thông báo cho Spring MVC đọc resouce bundle là messages:

@Bean("messageSource")  
public MessageSource messageSource() {  
 ReloadableResourceBundleMessageSource messageSource=new ReloadableResourceBundleMessageSource();  
 messageSource.setBasename("classpath:locale/messages");  
 messageSource.setDefaultEncoding("UTF-8");  
 messageSource.setUseCodeAsDefaultMessage(true);  
 return messageSource;  
}

**Báo cho Spring MVC sử dụng local nào**

Phương pháp phổ biến nhất để chọn ngôn ngữ sử dụng cho người dùng là bằng cách đọc giá trị accept-language từ trong số các headers của request gửi tới từ người dùng. Giá trị này mang thông tin về tùy chọn ngôn ngữ của người dùng.

Các phương pháp khác bao gồm đọc một giá trị nhất định từ session hoặc cookie.

Để chọn một locale trong Spring MVC, sử dụng bean localeResolver. Có một số triển khai có sẵn, bao gồm:

AcceptHeaderLocaleResolver

SessionLocaleResolver

CookieLocaleResolver

Tất cả đều là một phần của package org.springframework.web.servlet.i18n.

Đây là cấu hình bean localeResolver sử dụng CookieLocalResolver:

@Bean  
public LocaleResolver localeResolver() {  
 CookieLocaleResolver localeResolver = new CookieLocaleResolver();  
 return localeResolver;  
}

**Hiển thị các message**

Hiển thị message trong Thymeleaf sử dụng cú pháp như sau:

label th:text="#{greetings}"></label>

Tham khảo các cách thức truy cập message sử dụng Thymeleaf [tại đây](https://www.thymeleaf.org/doc/articles/springmvcaccessdata.html).

Last modified: Monday, 2 December 2019, 10:03 AM