

# **Bài 13**

# **Localization và Internationalization**

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT

# Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài "AJAX"  
Tóm tắt lại các phần đã học từ bài "AJAX"

- Trình bày được khái niệm và ý nghĩa của Internationalization
- Trình bày được khái niệm và ý nghĩa của Localization
- Sử dụng được MessageSource trong ứng dụng Spring
- Triển khai được i18n trong ứng dụng Spring
- Triển khai được i18n cho Controller
- Triển khai được i18n cho View
- Triển khai được i18n cho các thông báo có sử dụng tham số

# Thảo luận

I18N

Localization

ResourceBundle

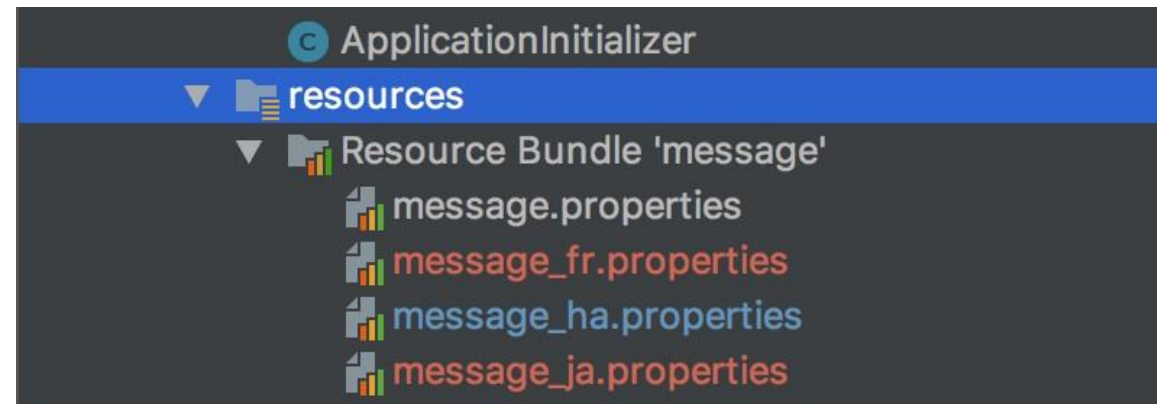
MessageSource

- Là quá trình thiết kế một ứng dụng phần mềm để nó có thể đáp ứng được nhiều ngôn ngữ và khu vực khác nhau mà không cần thay đổi kỹ thuật.
- Còn được gọi là **Translation** hay **Localization Enablement**
- Được gọi tắt là i18n, 18 là số chữ cái nằm giữa **I** và **N**

- Là quá trình điều chỉnh phần mềm đã được quốc tế hóa cho một ngôn ngữ hoặc một khu vực cụ thể bằng cách chỉ định ngôn ngữ hoặc khu vực sau đó dịch văn bản.
- Được gọi tắt là **l10n**, 10 là số chữ cái nằm giữa **L** và **N**
- Sự kết hợp giữa **i18n** và **l10n** đôi khi được gọi là **Globalization** và được gọi tắt là **g11n**
- Trong giao tiếp phổ thông, **i18n** được ngầm hiểu là **g11n**

# ResourceBundle

- Một tập các file properties
- Tuân theo quy ước đặt tên:  
    <bundle\_name>\_<locale>.properties
- Các message có thể có parameter
- Ví dụ  
    message=Alligators rock!  
    argument.required=The {0} argument is required



- Giao diện của pattern Strategy
- Được dùng để giải quyết message (lấy message từ source, theo yêu cầu)
- Có sẵn hai implements:
  - **ResourceBundleMessageSource**: được cung cấp như một ResourceBundle chuẩn.
  - **StaticMessageSource** : cung cấp những cách thức để lập trình bổ sung message vào nguồn.
- Hỗ trợ sẵn cho **i18n** và **L10n**



# Bean "messageSource"

- Là đối tượng của class triển khai interface MessageSource
- Được khai báo cho ApplicationContext
- Quá trình khai báo cần thông tin đầu vào là tên của các Resource Bundle

```
<bean id="messageSource"  
    class="org.springframework.context.support.ResourceBundleMessageSource">  
    <property name="basename" value="message"/>  
</bean>
```

```
@Bean  
public MessageSource messageSource() {  
    ResourceBundleMessageSource messageSource = new ResourceBundleMessageSource();  
    messageSource.setBasename("message");  
    return messageSource;  
}
```

# Các phương thức getMessage

- Cung cấp khả năng lấy message dựa theo **code**, **locale**, tham số (nếu có) của message, có thể có default message (trong trường hợp không tìm thấy) hoặc không
- **code**, **defaultMessage**, và tham số của message có thể được gói vào một đối tượng **MessageSourceResolvable**

```
String code, Object[] args, String defaultMessage, Locale locale  
String code, Object[] args, Locale locale  
MessageSourceResolvable resolvable, Locale locale
```

```
String loginTitleMessage = messageSource.getMessage();  
  
String franceLogMessage = messageSource.getMessage( code: "login.logged",  
    new Object[]{user.getUserName(), new Date()},  
    defaultMessage: "{0} logged in: {1}",  
    Locale.FRANCE);  
  
System.out.println(franceLogMessage); // sonnb logged in: 29/05/18 22:58
```

# Interceptor

- Được đính vào trong một phần vòng đời của request
- Có khả năng xử lý tiền/hậu kỳ các request
- Được dùng rất phổ biến để internationalize

```
@Override  
public void addInterceptors(InterceptorRegistry registry) {  
    LocaleChangeInterceptor interceptor = new LocaleChangeInterceptor();  
    interceptor.setParamName("lang");  
    registry.addInterceptor(interceptor);  
}
```

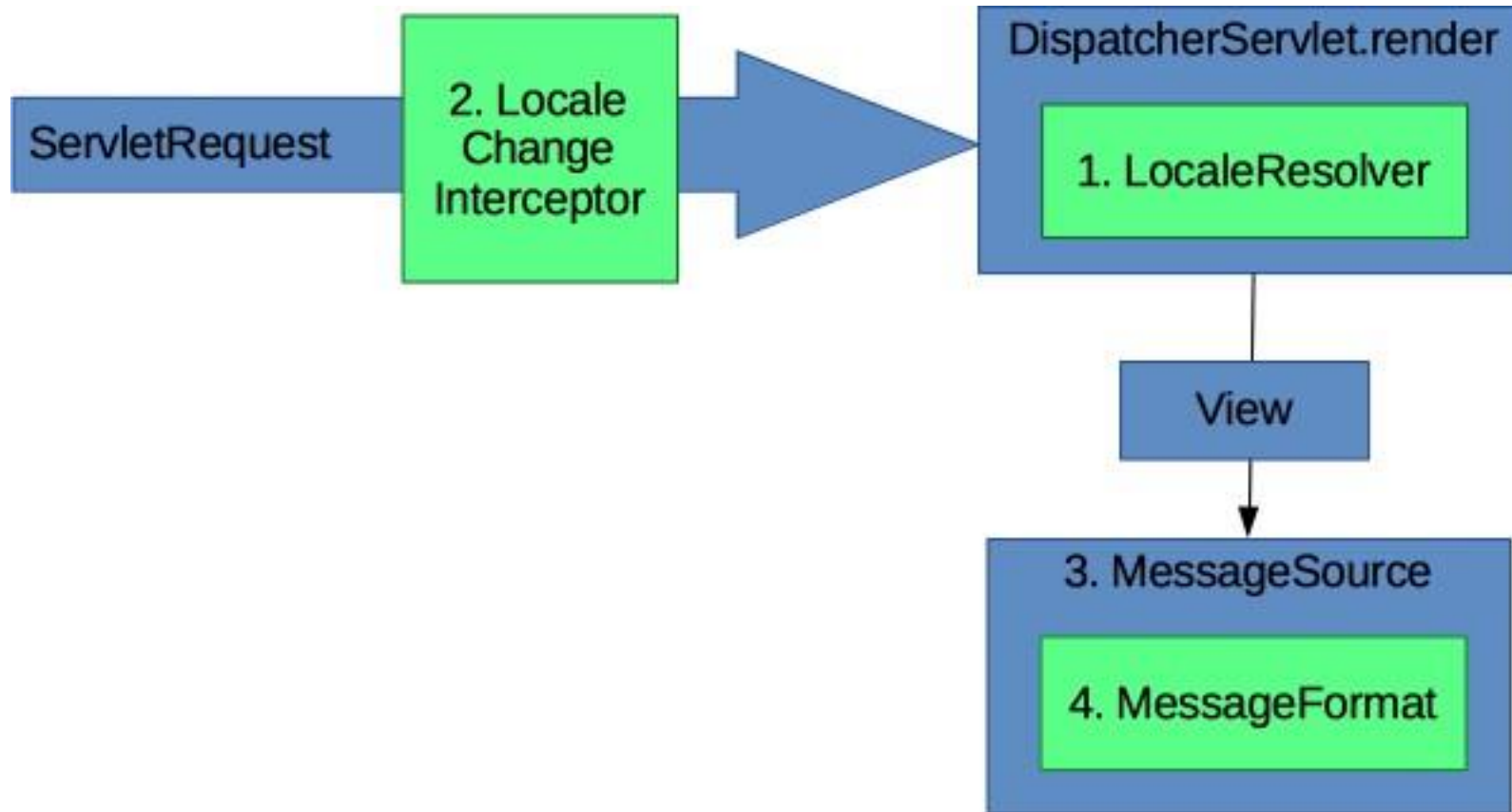
- Giao diện của strategy pattern, để triển khai phân giải message theo locale trên nền web.
- Có thể phân giải message dựa trên thông tin về locale trong request, session, cookie...

```
@Bean
public LocaleResolver localeResolver() {
    SessionLocaleResolver localeResolver = new SessionLocaleResolver();
    localeResolver.setDefaultLocale(new Locale(language: "en"));
    return localeResolver;
}
```

# Demo

I18N

# Tóm tắt bài học



# Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập  
Chuẩn bị bài tiếp theo: Spring Security