[Thực hành] Ứng dụng login

Mục tiêu

Luyện tập sử dụng spring message bundle, luyện tập triển khai i18n và l10n.

Mô tả

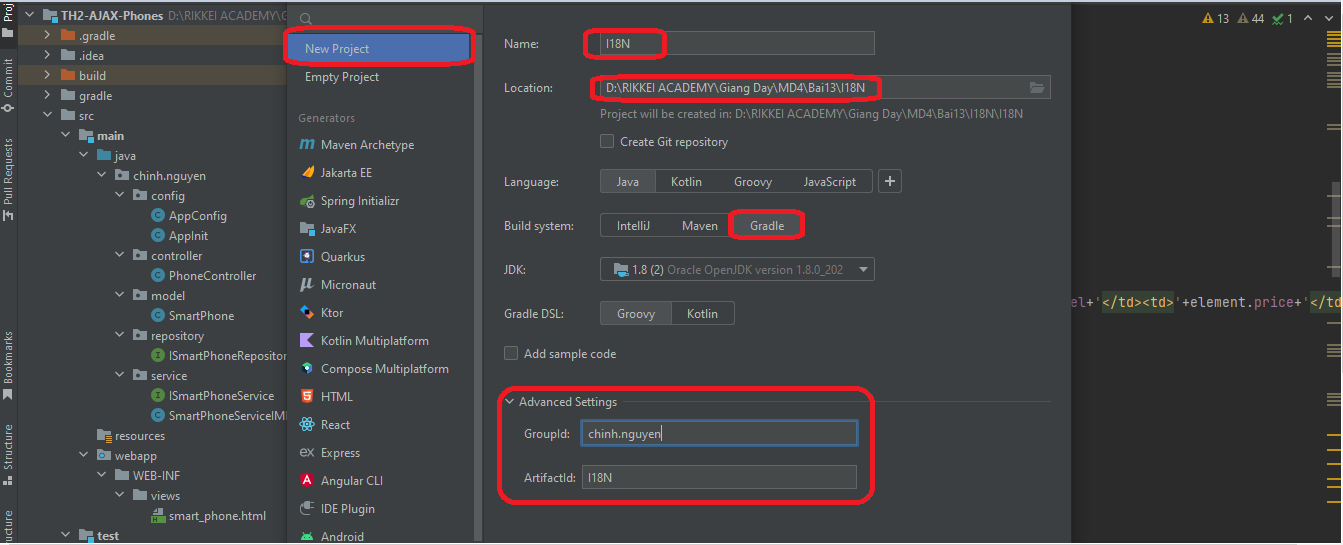
Với tài nguyên là một dự án Spring MVC đang chạy, triển khai internationalize.



Sau khi hoàn thành, học viên sẽ:

* Cấu hình được Spring sử dụng messageSource
* Sử dụng được message source
* Sử dụng được LocaleChangeInterceptor để hỗ trợ localization
* Sử dụng được SessionLocaleResolver để hỗ trợ localization

Hướng dẫn



* Vào build.gradle import các thư viện:

compileOnly('javax.servlet:javax.servlet-api:4.0.1')  
implementation group: 'org.springframework', name: 'spring-core', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-context', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-beans', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-web', version: '5.3.2'  
implementation group: 'org.springframework', name: 'spring-webmvc', version: '5.3.2'  
implementation group: 'org.thymeleaf', name: 'thymeleaf-spring5', version: '3.0.11.RELEASE'  
implementation group: 'nz.net.ultraq.thymeleaf', name: 'thymeleaf-layout-dialect', version: '2.5.2'  
implementation group: 'org.springframework.data', name: 'spring-data-jpa', version: '2.4.2'  
implementation group: 'org.springframework', name: 'spring-tx', version: '5.3.2'

plugins **{** id 'java'  
 id 'war'  
**}**

* Tạo package: **chinh.nguyen.config**
* Tạo class: **AppConfig** trong package: **config**

package chinh.nguyen.config;  
  
import org.springframework.beans.BeansException;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.ApplicationContextAware;  
import org.springframework.context.MessageSource;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.support.ResourceBundleMessageSource;  
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;  
import org.springframework.data.web.config.EnableSpringDataWebSupport;  
import org.springframework.transaction.annotation.EnableTransactionManagement;  
import org.springframework.web.servlet.LocaleResolver;  
import org.springframework.web.servlet.config.annotation.EnableWebMvc;  
import org.springframework.web.servlet.config.annotation.InterceptorRegistry;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
import org.springframework.web.servlet.i18n.CookieLocaleResolver;  
import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;  
import org.springframework.web.servlet.i18n.SessionLocaleResolver;  
import org.thymeleaf.spring5.SpringTemplateEngine;  
import org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver;  
import org.thymeleaf.spring5.view.ThymeleafViewResolver;  
import org.thymeleaf.templatemode.TemplateMode;  
  
import java.util.Locale;  
  
@Configuration  
@EnableWebMvc  
@EnableTransactionManagement  
@EnableSpringDataWebSupport  
@ComponentScan("chinh.nguyen.controller")  
public class AppConfig implements WebMvcConfigurer, ApplicationContextAware {  
 private ApplicationContext applicationContext;  
  
 @Override  
 public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {  
 this.applicationContext = applicationContext;  
 }  
  
 //Cấu hình Thymleaf  
 @Bean  
 public SpringResourceTemplateResolver templateResolver() {  
 SpringResourceTemplateResolver templateResolver = new SpringResourceTemplateResolver();  
 templateResolver.setApplicationContext(applicationContext);  
 templateResolver.setPrefix("/WEB-INF/views");  
 templateResolver.setSuffix(".html");  
 templateResolver.setTemplateMode(TemplateMode.*HTML*);  
 templateResolver.setCharacterEncoding("utf-8");  
 return templateResolver;  
 }  
  
 @Bean  
 public SpringTemplateEngine templateEngine() {  
 SpringTemplateEngine templateEngine = new SpringTemplateEngine();  
 templateEngine.setTemplateResolver(templateResolver());  
 templateEngine.setMessageSource(messageSource());  
 return templateEngine;  
 }  
  
 @Bean  
 public ThymeleafViewResolver viewResolver() {  
 ThymeleafViewResolver viewResolver = new ThymeleafViewResolver();  
 viewResolver.setTemplateEngine(templateEngine());  
 viewResolver.setCharacterEncoding("utf-8");  
 viewResolver.setContentType("utf-8");  
 return viewResolver;  
 }  
 //Config resources  
 @Bean  
 public MessageSource messageSource() {  
 ResourceBundleMessageSource messageSource = new ResourceBundleMessageSource();  
 messageSource.setBasename("message");  
 messageSource.setDefaultEncoding("utf-8");  
 return messageSource;  
 }

}

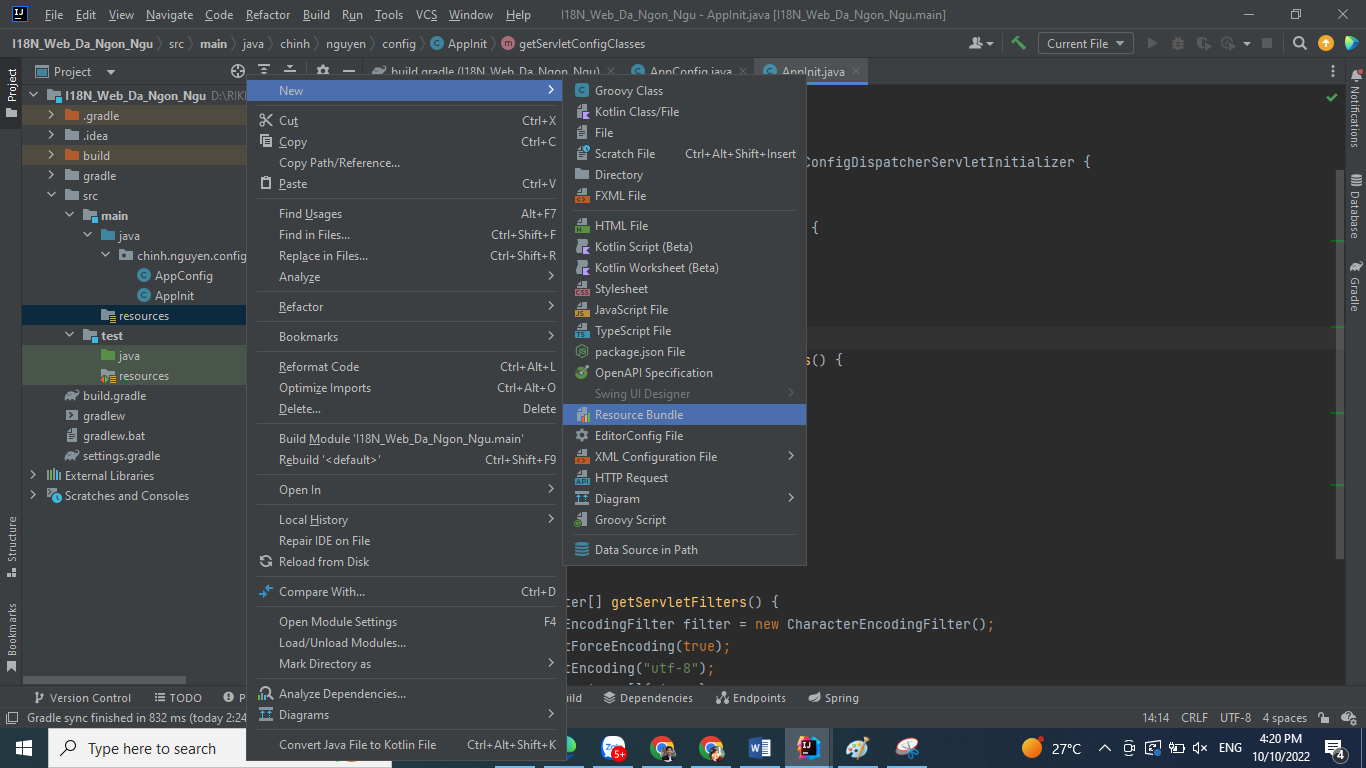
* Tạo class: **AppInit:**

package chinh.nguyen.config;  
  
import org.springframework.web.filter.CharacterEncodingFilter;  
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;  
  
import javax.servlet.Filter;  
  
public class AppInit extends AbstractAnnotationConfigDispatcherServletInitializer {  
 @Override  
 protected Class<?>[] getRootConfigClasses() {  
 return new Class[]{AppConfig.class};  
 }  
  
 @Override  
 protected Class<?>[] getServletConfigClasses() {  
 return new Class[0];  
 }  
  
 @Override  
 protected String[] getServletMappings() {  
 return new String[]{"/"};  
 }  
 @Override  
 protected Filter[] getServletFilters() {  
 CharacterEncodingFilter filter = new CharacterEncodingFilter();  
 filter.setForceEncoding(true);  
 filter.setEncoding("utf-8");  
 return new Filter[]{filter};  
 }  
}

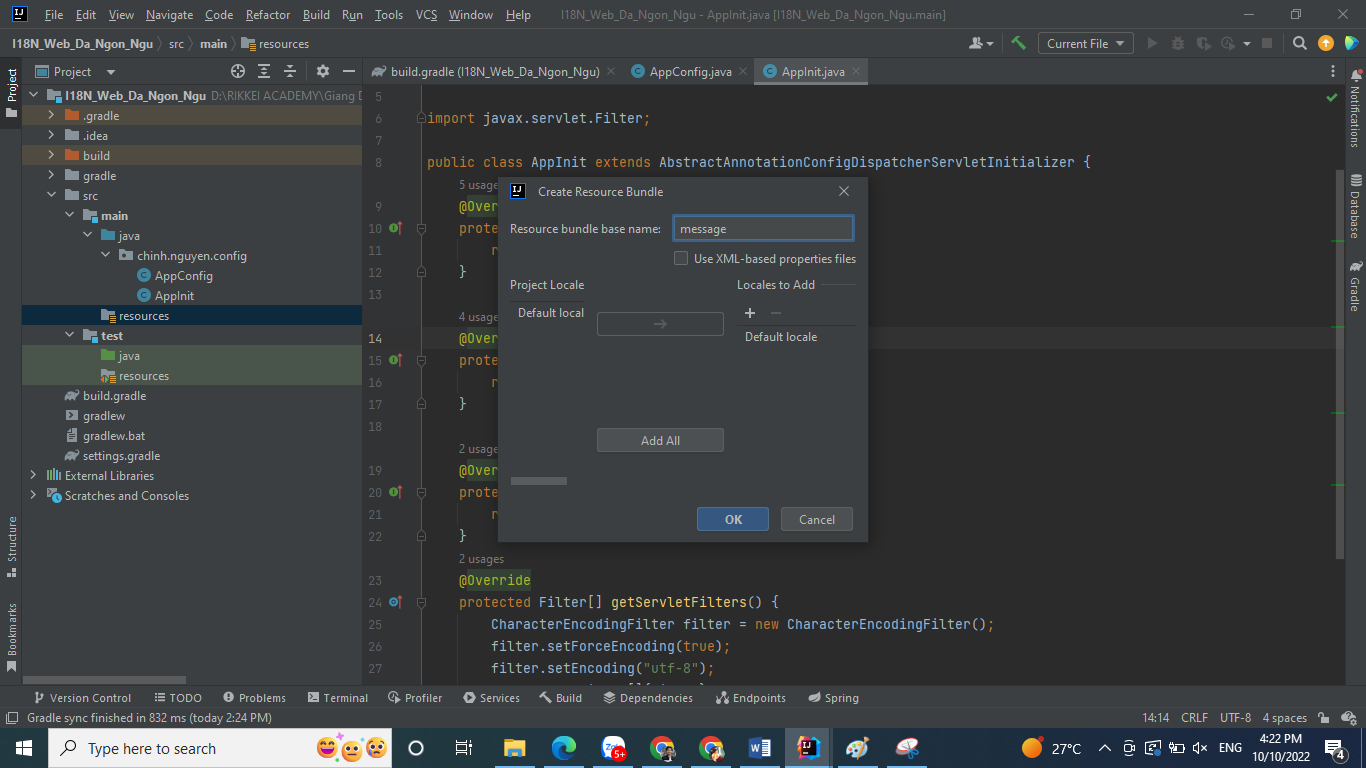
* Tạo package: **model** trong package: **chinh.nguyen** => Tạo class: **User** trong package này:

private Long id;  
private String username;  
private String password;  
  
public User() {  
}  
  
public User(Long id, String username, String password) {  
 this.id = id;  
 this.username = username;  
 this.password = password;  
}  
  
public Long getId() {  
 return id;  
}  
  
public void setId(Long id) {  
 this.id = id;  
}  
  
public String getUsername() {  
 return username;  
}  
  
public void setUsername(String username) {  
 this.username = username;  
}  
  
public String getPassword() {  
 return password;  
}  
  
public void setPassword(String password) {  
 this.password = password;  
}

* Tạo package: **controller** => Tạo class: **AuthController** trong package này:
* Tạo file: **message.properties** từ thư mục **resources**:



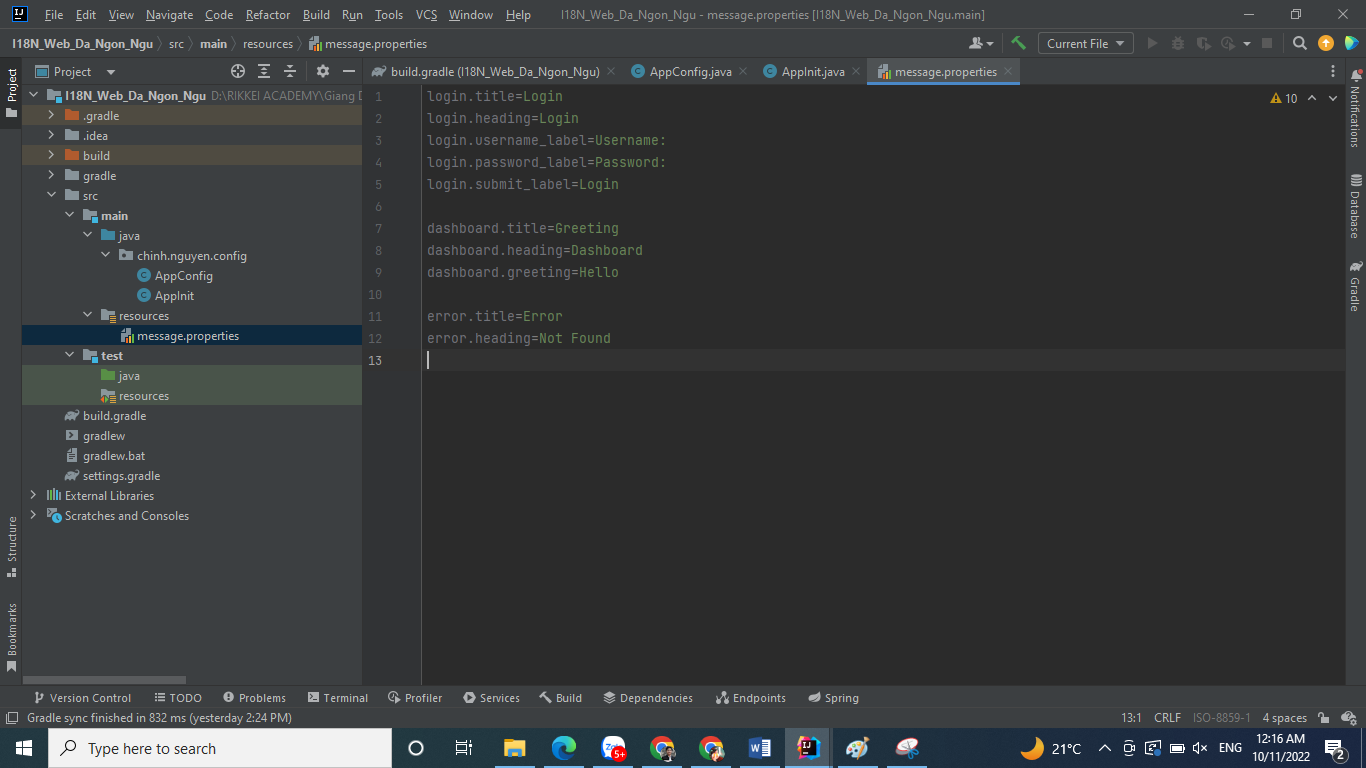
* Đặt tên **message**:



* Thêm đoạn code sau vào file message.properties:

login.title=Login  
login.heading=Login  
login.username\_label=Username:  
login.password\_label=Password:  
login.submit\_label=Login  
  
dashboard.title=Greeting  
dashboard.heading=Dashboard  
dashboard.greeting=Hello  
  
error.title=Error  
error.heading=Not Found

* Cấu trúc dự án lúc này:



* Khai báo cấu hình bean messageSource, ở đây là mã mẫu khi khai báo bằng annotation trong file cấu hình bằng java.

@Bean  
public MessageSource messageSource() {  
 ResourceBundleMessageSource messageSource = new ResourceBundleMessageSource();  
 messageSource.setBasename("message");  
 messageSource.setDefaultEncoding("UTF-8");  
 return messageSource;  
}

Trong đó, giá trị "message" chính là tên file trong file message.properties.

Nhớ thêm các imports cần thiết.

Sau đó, ở template, gọi tới tên của các message trong file **message.properties** thay vì sử dụng hard-coded, sau đây là mã tham khảo khi template là html và được resolve bằng thư viện thymeleaf, chú ý cú pháp **#{messagePath}** trong file **login.html** trong thư mục **webapp/WEB-INF/views**

<!doctype html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta http-equiv='Content-Type' content='text/html; charset=UTF-8' />  
 <title th:text="#{login.title}"></title>  
</head>  
<body>  
<h1 th:text="#{login.heading}"></h1>  
Language: <a href="?lang=en">English</a> | <a>Hawaii</a>  
<form th:action="@{/doLogin}" th:object="${user}" method="post">  
 <table>  
 <tr>  
 <td th:text="#{login.username\_label}"></td>  
 <td><input type="text" th:field="\*{username}"/></td>  
 </tr>  
 <tr>  
 <td th:text="#{login.password\_label}"></td>  
 <td><input type="password" th:field="\*{password}"/></td>  
 </tr>  
 <tr>  
 <td></td>  
 <td><input type="submit" th:value="#{login.submit\_label}"></td>  
 </tr>  
 </table>  
</form>  
</body>  
</html>

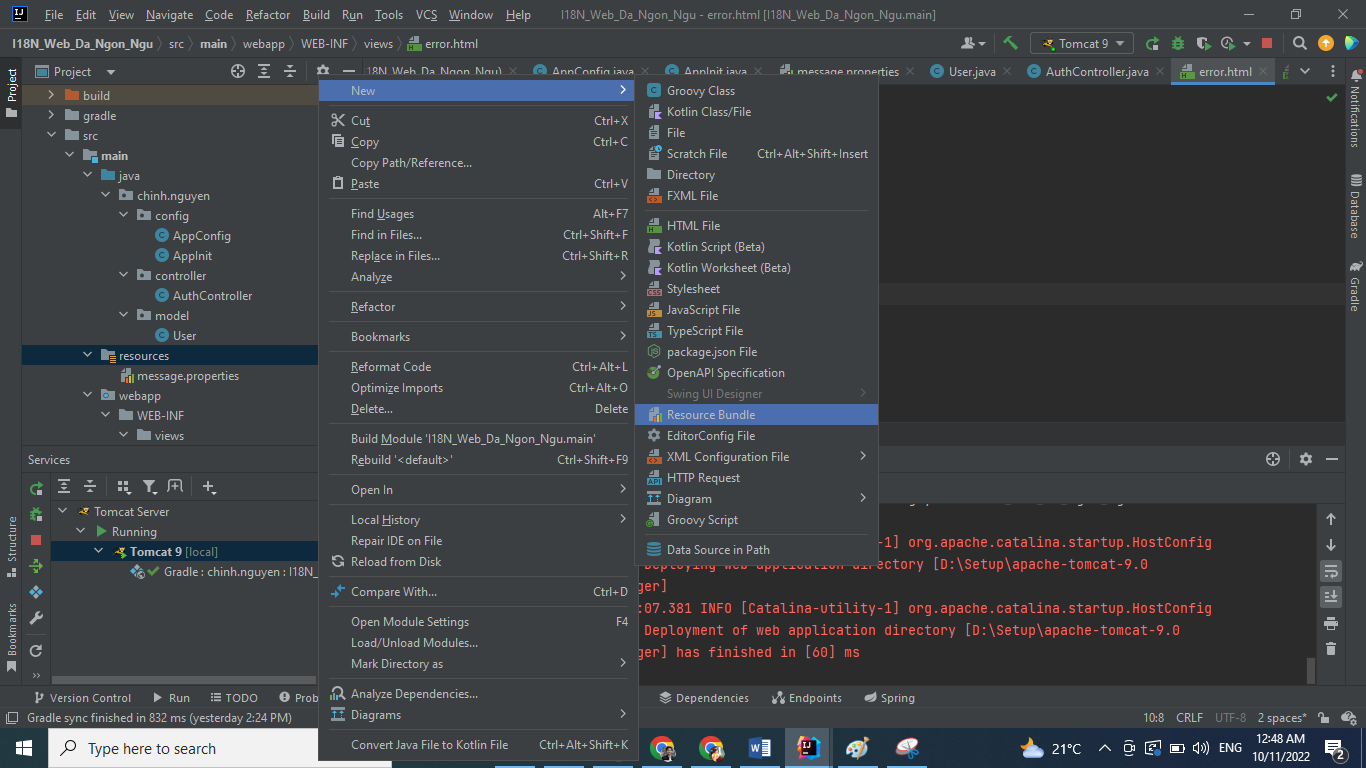
* Tạo file: **success.html** trong folder **views**:

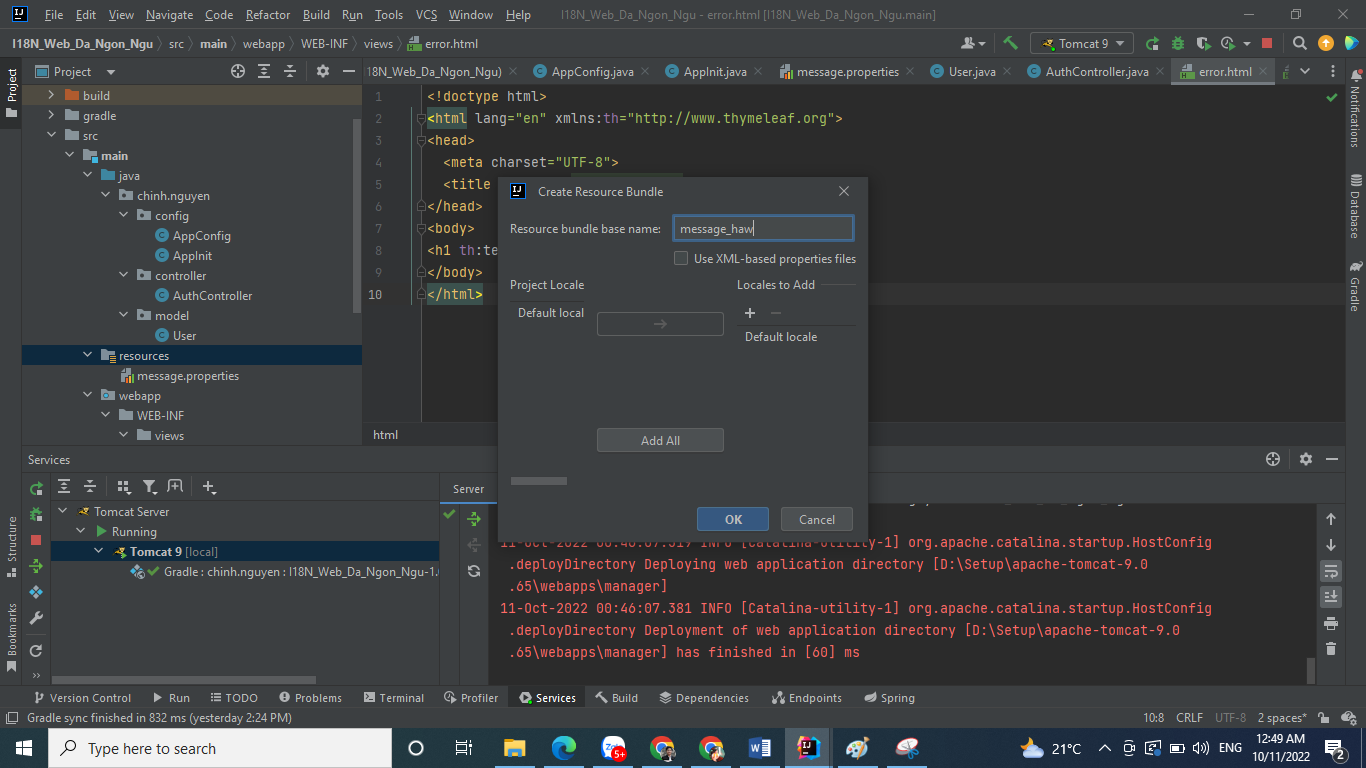
<!doctype html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title th:text="#{dashboard.title}"></title>  
</head>  
<body>  
<h1 th:text="#{dashboard.heading}"></h1>  
<span th:text="#{dashboard.greeting}"></span> <span th:text="${user.username}"></span>  
</body>  
</html>

* Tạo file: **error.html** trong folder **views**:

<!doctype html>  
<html lang="en" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title th:text="#{error.title}"></title>  
</head>  
<body>  
<h1 th:text="#{error.heading}"></h1>  
</body>  
</html>

* Add Tomcat chạy chương trình, truy cập vào API: <http://localhost:8080/login>
* Đăng nhập thử với: username: **admin** pass: **123456**





login.title=E komo i ka papa uila  
login.heading=E komo i ka papa uila  
login.username\_label=Inoa inoa:  
login.password\_label=Hua huna:  
login.submit\_label=Komo mai  
  
dashboard.title=Aloha  
dashboard.heading=?O ka papa  
dashboard.greeting=Aloha  
  
error.title=Hewa  
error.heading=Loa?a 'ole

Trong file cấu hình **AppConfig**, cấu hình để bổ sung LocaleChangeInterceptor.

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

Interceptor này sẽ tự động phân tích tham số **lang** đi kèm các request để xác định bản địa hiện tại.

Tham số lang được định nghĩa ở file **login.html:**

Language: <a href="?lang=en">English</a> | <a>Hawaii</a>

Tiếp theo, bổ sung bean localeResolver để sử dụng thông tin về bản địa và sử dụng message bundle tương ứng:

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

Ở đây, ta đã khai báo một localeResolver lấy thông tin về bản địa từ session của người dùng, đồng thời dự phòng sử dụng ngôn ngữ là “en”.

Bước tiếp theo, trong template, tạo một thẻ anchor đơn giản để gửi request với tham số “lang”:

* Cập nhật lại tham số **lang** trong file: **login.html:**

Language: <a href="?lang=en">English</a> | <a href="?lang=haw">Hawaii</a>

#### Chạy lại chương trình tích sang ngôn ngữ Hawaii kiểm tra kết quả:



#### **Bản địa hóa dựa trên session**

Hãy tiếp tục bản địa hóa trang dashboard hay một trang khác trong project:

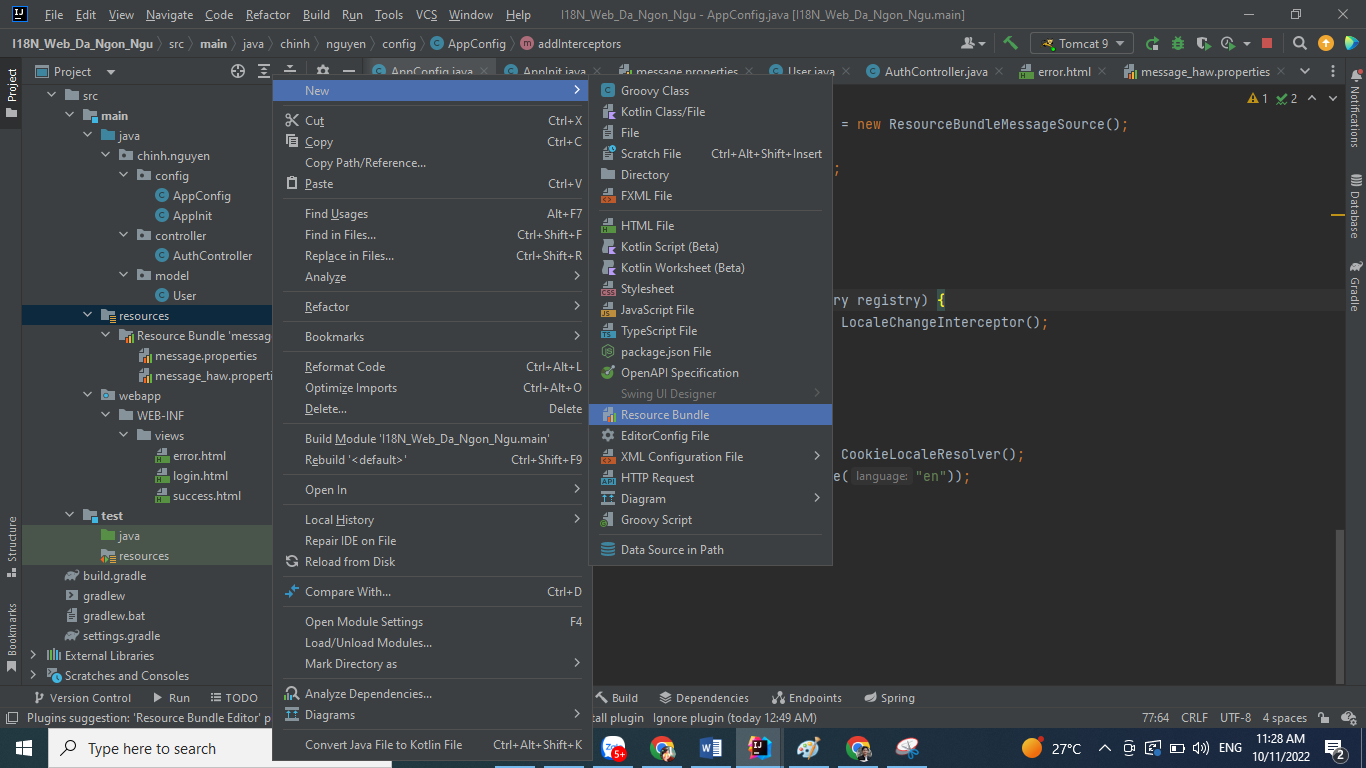
Sau đó kiểm tra lại để thấy rằng việc bản địa hóa vẫn tiếp tục được thực hiện mà không cần tới tham số lang trên tất cả các request. Đó là vì ở đây đang sử dụng LocaleResolver là một SessionLocaleResolver.

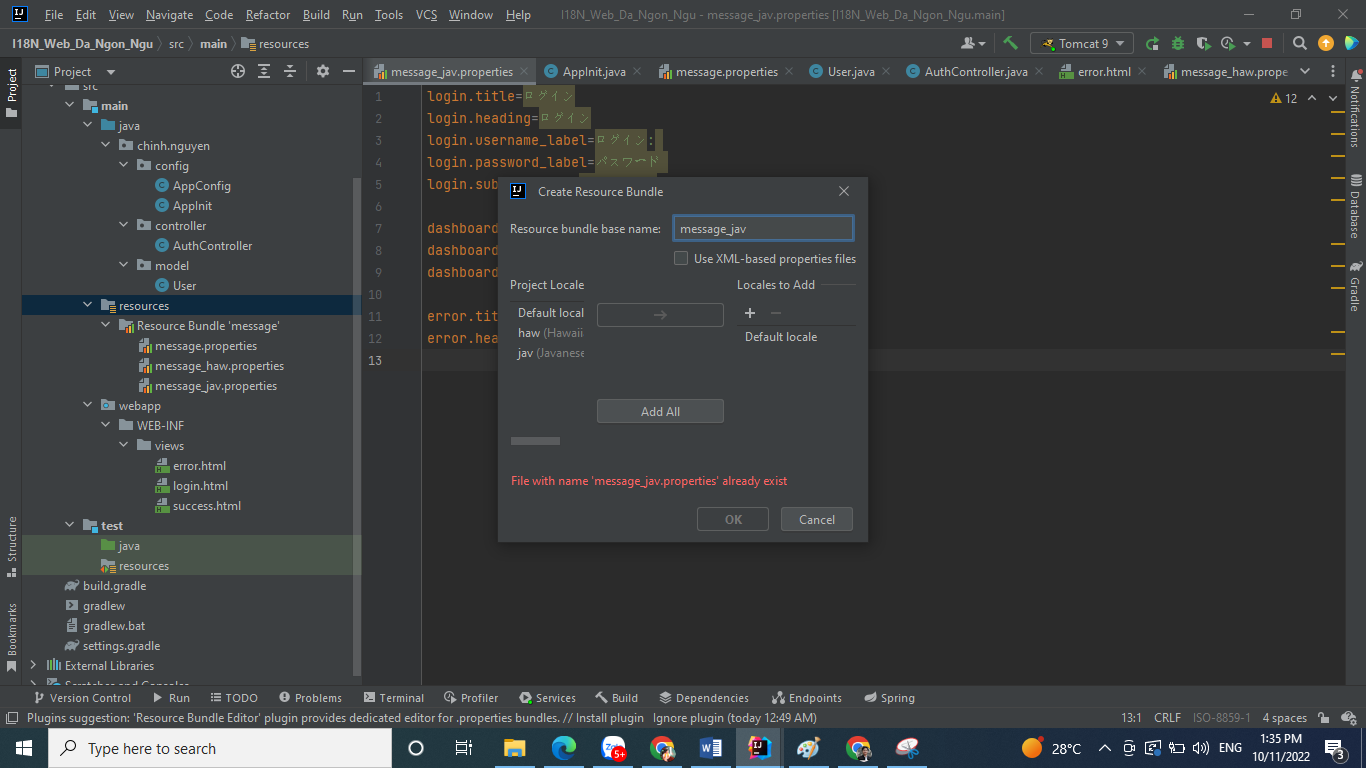
Bản địa hóa dựa trên **cookie:**

* Bây giờ chúng ta sẽ sửa một chút đoạn code trong **AppConfig** từ session sang cookie

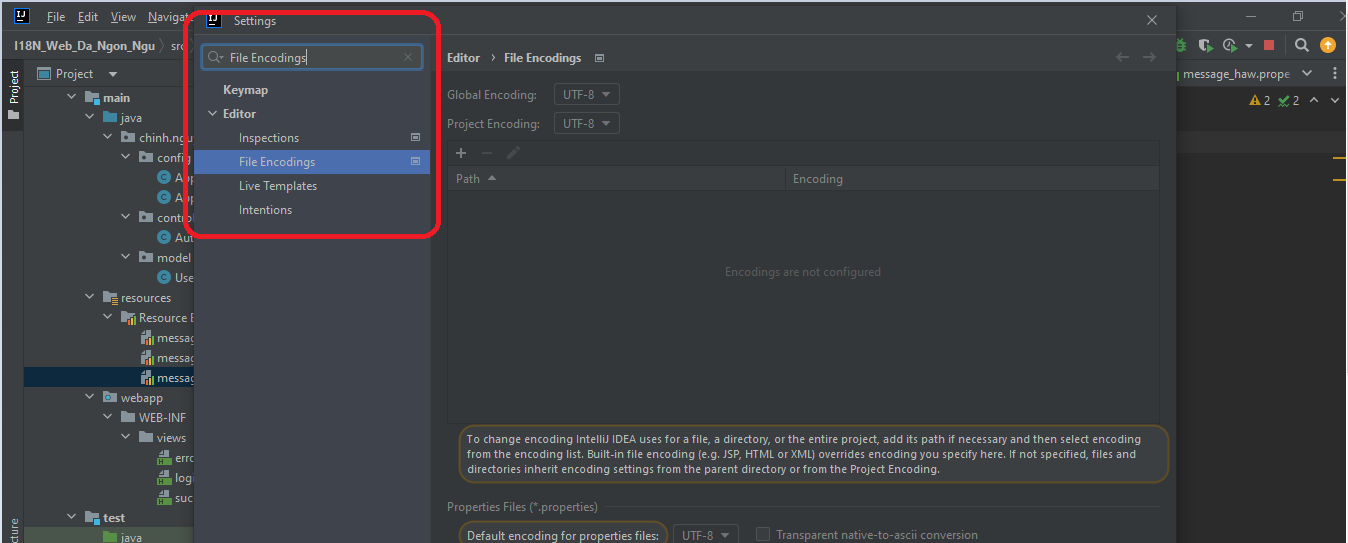
@Bean  
public LocaleResolver localeResolver() {  
 CookieLocaleResolver localeResolver = new CookieLocaleResolver();  
 localeResolver.setDefaultLocale(new Locale("en"));  
 return localeResolver;  
}

* Thêm một bản địa hóa với ngôn ngữ JAV.





* Để chuẩn bản địa sang các ngôn ngữ có kiểu UTF-8 có thể chúng ta sẽ phải xử lý encode bằng cách như sau: File => Settings
* Search File Encodings như hình:



* Chỉnh sửa toàn bộ sang UTF-8 rồi Apply => OK

Thêm code vào file: **message\_jav.properties**

login.title=ログイン  
login.heading=ログイン  
login.username\_label=ログイン:   
login.password\_label=パスワード   
login.submit\_label=ログインする  
  
dashboard.title=こんにちは  
dashboard.heading=ダッシュボード  
dashboard.greeting=こんにちは  
  
error.title=エラー  
error.heading=見つかりません

* Chỉnh sửa lại file: **login.html**

Language: <a href="?lang=en">English</a> | <a href="?lang=haw">Hawaii</a>> | <a href="?lang=jav">JAV</a>



### Hướng dẫn nộp bài:

Up code lên github.

<https://github.com/nguyendongminhtri/MD4-Bai13-TH-I18N.git>

Paste link github vào phần nộp bài.