package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication

@ComponentScan("com.\*")

public class TmsApplication {

/\*\*

\* Starting point of the application

\*

\* @param args Arguments passed to the application

\*/

public static void main(String[] args) {

SpringApplication.run(TmsApplication.class, args);

}

}

package com.controller;

import java.util.Locale;

import org.springframework.context.MessageSource;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.support.ReloadableResourceBundleMessageSource;

import org.springframework.validation.beanvalidation.LocalValidatorFactoryBean;

import org.springframework.web.servlet.LocaleResolver;

import org.springframework.web.servlet.config.annotation.InterceptorRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

import org.springframework.web.servlet.i18n.LocaleChangeInterceptor;

import org.springframework.web.servlet.i18n.SessionLocaleResolver;

@Configuration

public class InternationalizationConfig extends WebMvcConfigurerAdapter {

/\*\*

\* Set default Locale

\*

\* @return A bean of LocalResolver

\*/

@Bean

public LocaleResolver localeResolver() {

SessionLocaleResolver slr = new SessionLocaleResolver();

slr.setDefaultLocale(Locale.US);

return slr;

}

/\*\*

\* Set path variable name for changing language

\*

\* @return A bean of LocaleChangeInterceptor

\*/

@Bean

public LocaleChangeInterceptor localeChangeInterceptor() {

LocaleChangeInterceptor lci = new LocaleChangeInterceptor();

lci.setParamName("language");

return lci;

}

/\*\*

\* Add interceptor into the registry

\*/

@Override

public void addInterceptors(InterceptorRegistry registry) {

registry.addInterceptor(localeChangeInterceptor());

}

/\*\*

\* Set base name for messages.properties files Set default encoding to UTF-8

\*

\* @return A bean of MessageSource

\*/

@Bean

public MessageSource messageSource() {

ReloadableResourceBundleMessageSource rrbms = new ReloadableResourceBundleMessageSource();

rrbms.setBasename("classpath:messages");

rrbms.setDefaultEncoding("UTF-8");

return rrbms;

}

/\*\*

\* Set validation message source

\*

\* @return A bean of LocalValidatorFactoryBean

\*/

@Bean

public LocalValidatorFactoryBean localValidatorFactoryBean() {

LocalValidatorFactoryBean lvfb = new LocalValidatorFactoryBean();

lvfb.setValidationMessageSource(messageSource());

return lvfb;

}

}

package com.controller;

import java.util.Arrays;

import java.util.List;

import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import com.model.UserClaim;

import com.service.TaxService;

@Controller

public class TaxController {

@Autowired

public TaxService taxService;

/\*\*

\* Display taxclaim.jsp page when a get request is pushed on url

\* /getTaxClaimFormPage

\*

\* @param userClaim Is the UserClaim component

\* @return taxclaim as a jsp page

\* @see UserClaim

\*/

@RequestMapping(value = "/getTaxClaimFormPage", method = RequestMethod.GET)

public String claimPage(@ModelAttribute("userClaim") UserClaim userClaim) {

return "taxclaim";

}

/\*\*

\* Return result.jsp age when validation is successful Otherwise return back to

\* taxclaim page with error message

\*

\* @param userClaim UserClaim component

\* @param result BindingResult which validate the user input

\* @param map ModelMap to put attribute which will be forwarded to next

\* page

\* @return "result.jsp" page if the validation is successful otherwise

\* "taxclaim.jsp" with error included

\*/

@RequestMapping(value = "/calculateTax", method = RequestMethod.GET)

public String calculateTax(@Valid @ModelAttribute("userClaim") UserClaim userClaim, BindingResult result,

ModelMap map) {

if (result.hasErrors()) {

return "taxclaim";

}

double amount = taxService.calculateTax(userClaim);

map.addAttribute("amount", amount);

return "result";

}

/\*\*

\* Populate <form:select /> tag in the taxclaim.jsp page

\*

\* @return List of expenses

\*/

@ModelAttribute("expenseList")

public List<String> populateExpense() {

return Arrays.asList("MedicalExpense", "TravelExpense", "FoodExpense");

}

}

package com.model;

import javax.validation.constraints.NotBlank;

import javax.validation.constraints.PositiveOrZero;

import javax.validation.constraints.Size;

import org.springframework.stereotype.Component;

@Component

public class UserClaim {

private String expenseType;

@PositiveOrZero(message = "{error.expenseAmount.negative}")

private double expenseAmt;

@NotBlank(message = "{error.employeeId}")

@Size(min = 5, message = "{error.employeeId.size}")

private String employeeId;

public String getExpenseType() {

return expenseType;

}

public void setExpenseType(String expenseType) {

this.expenseType = expenseType;

}

public double getExpenseAmt() {

return expenseAmt;

}

public void setExpenseAmt(double expenseAmt) {

this.expenseAmt = expenseAmt;

}

public String getEmployeeId() {

return employeeId;

}

public void setEmployeeId(String employeeId) {

this.employeeId = employeeId;

}

}

package com.service;

import org.springframework.stereotype.Service;

import com.model.UserClaim;

@Service

public interface TaxService {

/\*\*

\* Calculate Tax

\*

\* @param userClaim UserClaim bean

\* @return Calculated tax

\*/

public double calculateTax(UserClaim userClaim);

}

package com.service;

import org.springframework.stereotype.Service;

import com.model.UserClaim;

@Service

public class TaxServiceImpl implements TaxService {

/\*\*

\* Calculate the tax according to the srs

\*

\* @param userClaim UserClaim component to get the values

\* @return Calculated tax

\*/

@Override

public double calculateTax(UserClaim userClaim) {

String e = userClaim.getExpenseType();

double a = userClaim.getExpenseAmt();

double t = 0.0;

if (e.startsWith("M")) {

if (a <= 1000) {

t = 15.0;

} else if (a > 1000 && a <= 10000) {

t = 20.0;

} else if (a > 10000) {

t = 25.0;

}

} else if (e.startsWith("T")) {

if (a <= 1000) {

t = 10.0;

} else if (a > 1000 && a <= 10000) {

t = 15.0;

} else if (a > 10000) {

t = 20.0;

}

} else if (e.startsWith("F")) {

if (a <= 1000) {

t = 5.0;

} else if (a > 1000 && a <= 10000) {

t = 10.0;

} else if (a > 10000) {

t = 15.0;

}

}

return a \* (t / 100.0);

}

}

server.port=9095

spring.mvc.view.prefix=/WEB-INF/jsp/

spring.mvc.view.suffix=.jsp

spring.mvc.static-class-path=/resources/\*\*

---------------de.pro

label.employeeId=Employee ID in German

label.expenseType=Expense Type in German

label.expenseAmount=Expense Amount in German

error.employeeId=Employee ID cannot be empty in German

error.employeeId.size=Employee ID should be at least 5 characters in German

error.expenseAmount=Expense Amount cannot be empty in German

error.expenseAmount.numeric=Expense amount should be numeric only in German

error.expenseAmount.negative=Expense amount should not be a negative number in German

------ fr.pro

label.employeeId=Employee ID in French

label.expenseType=Expense Type in French

label.expenseAmount=Expense Amount in French

error.employeeId=Employee ID cannot be empty in French

error.employeeId.size=Employee ID should be at least 5 characters in French

error.expenseAmount=Expense Amount cannot be empty in French

error.expenseAmount.numeric=Expense amount should be numeric only in French

error.expenseAmount.negative=Expense amount should not be a negative number in French

------------- msg.pro

label.employeeId=Employee ID in English

label.expenseType=Expense Type in English

label.expenseAmount=Expense Amount in English

error.employeeId=Employee ID cannot be empty in English

error.employeeId.size=Employee ID should be at least 5 characters in English

error.expenseAmount=Expense Amount cannot be empty in English

error.expenseAmount.numeric=Expense amount should be numeric only in English

error.expenseAmount.negative=Expense amount should not be a negative number in English

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h2>The tax claim for ${ userClaim.expenseType } with expense amount

${ userClaim.expenseAmt } is ${ amount }</h2>

</body>

</html>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"* isELIgnored=*"false"*%>

<%@ taglib prefix=*"spring"* uri=*"http://www.springframework.org/tags"*%>

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<%@ taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<body style="background-color: *lavender*">

<h1>

<center>Tax: Tax Claim</center>

</h1>

<a href=*"/getTaxClaimFormPage?language=en"*>English</a>|

<a href=*"/getTaxClaimFormPage?language=de"*>German</a>|

<a href=*"/getTaxClaimFormPage?language=fr"*>French</a>

</align>

<form:form action=*"/calculateTax"* method=*"get"* modelAttribute=*"userClaim"*>

<table>

<tr>

<td id=*"id1"*>

<spring:message code=*"label.employeeId"* />

</td>

<td id=*"id2"*>

<form:input path=*"employeeId"* id=*"employeeId"* />

</td>

<td id=*"id3"*>

<form:errors path=*"employeeId"* />

</td>

</tr>

<tr>

<td id=*"id4"*>

<spring:message code=*"label.expenseType"* />

</td>

<td id=*"id5"*>

<form:select path=*"expenseType"* items=*"*${ expenseList }*"* id=*"expenseType"* />

</td>

<td id=*"id6"*></td>

</tr>

<tr>

<td id=*"id7"*>

<spring:message code=*"label.expenseAmount"* />

</td>

<td id=*"id8"*>

<form:input path=*"expenseAmt"* id=*"expenseAmount"* />

</td>

<td id=*id9*>

<form:errors path=*"expenseAmt"* />

</td>

</tr>

<tr>

<td><input type=*"Submit"* name=*"submit"* value=*"Calculate Claim"* /></td>

<td></td>

</tr>

<tr>

<td><input type=*"reset"* name=*"reset"* value=*"Clear"* /></td>

<td></td>

</tr>

</table>

</form:form>

</body>

</html>