Message 처리

|  |
| --- |
| <bean id=*"messageSourceAccessor"*  class=*"org.springframework.context.support.MessageSourceAccessor"*>  <constructor-arg ref=*"messageSource"* />  </bean>    <bean id=*"messageSource"* class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>  <property name=*"basenames"*>  <list>  <value>classpath:/message/message-common</value>  <value>classpath:/egovframework/rte/fdl/idgnr/messages/idgnr</value>  <value>classpath:/egovframework/rte/fdl/property/messages/properties</value>  </list>  </property>  <property name=*"cacheSeconds"*>  <value>60</value>  </property>  </bean>  <!-- spring messageSource -->  <!-- <bean id="messageSource\_spring" -->  <!-- class="org.springframework.context.support.ResourceBundleMessageSource"> -->  <!-- <property name="basenames"> -->  <!-- <list> -->  <!-- <value>message/message-common</value> -->  <!-- </list> -->  <!-- </property> -->  <!-- <property name="defaultEncoding" value="UTF-8" /> -->  <!-- </bean> --> |
| MessageSourceAccessor message = ApplicationUtil.*getWebApplicationContext*().getBean(MessageSourceAccessor.**class**); String msg = message.getMessage(atError.getCode()); |

Validator

|  |
| --- |
| JSR303 및 여러 개 걸려있는 것.  **public** **class** ValidationUtil {  **public** **static** Errors newErrors(Object object) {  **return** *newErrors*(object,object.getClass().getName());  }  **public** **static** Errors newErrors(Object object, String name) {  BeanPropertyBindingResult errors = **new** BeanPropertyBindingResult(object, name);  **return** errors;  }        /\* 스프링 validator  <bean id="validator" class="org.springframework.validation.beanvalidation.LocalValidatorFactoryBean" />  @Resource  Validator validator;  \*/  **public** **static** Errors validate(Validator validator,Object object){  **return** *validate*(validator,object,*newErrors*(object, object.getClass().getName()));  }  **public** **static** Errors validate(Validator validator,Object object ,Errors errors){  validator.validate(object, errors);  **return** errors;  }  } |

Validator

|  |
| --- |
| <bean id=*"messageSource"* class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>  <property name=*"basenames"*>  <list>  <value>classpath:/message/message-common</value>  <value>classpath:/egovframework/rte/fdl/idgnr/messages/idgnr</value>  <value>classpath:/egovframework/rte/fdl/property/messages/properties</value>  </list>  </property>  <property name=*"cacheSeconds"*>  <value>60</value>  </property>  </bean> |
| <%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"* %>  <%@ taglib prefix=*"validator"* uri=*"http://www.springmodules.org/tags/commons-validator"* %>  Jsp  <!DOCTYPE html>  <html lang=*"en-us"* class=*"no-js"*>  <head>  <script type=*"text/javascript"*>  <validator:javascript dynamicJavascript="false" staticJavascript="true"/>  </script>  <validator:javascript formName=*"sampleVO"* staticJavascript=*"false"* xhtml=*"true"* cdata=*"false"*/>  <meta charset=*"utf-8"*>  <title>Omnifit</title>  </head> |
| <!-- egov validator -->  <bean id=*"beanValidator"* class=*"org.springmodules.validation.commons.DefaultBeanValidator"*>  <property name=*"validatorFactory"* ref=*"validatorFactory"*/>  </bean>  <bean id=*"validatorFactory"* class=*"org.springmodules.validation.commons.DefaultValidatorFactory"*>  <property name=*"validationConfigLocations"*>  <list>  <value>/WEB-INF/config/validator/validator-rules.xml</value>  <value>/WEB-INF/config/validator/validator.xml</value>  </list>  </property>  </bean> |

Form error

|  |
| --- |
| <form:form commandName=*"FromToVO"*>  <div class=*"container"*>  <div class=*"row"*>  <%-- <form> --%>  <%-- <input type="text" name="from"/>[<form:errors path="from"/>] --%>  <input type=*"text"* name=*"from"*/>[<form:errors path=*"from"*/>]  <input type=*"text"* name=*"to"*/>[<form:errors path=*"to"*/>]  <input type=*"submit"* value=*"submit"*/>  <%-- </form> --%>  dddddddd  </div>  </div> |

Security

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:sec=*"http://www.springframework.org/schema/security"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://www.springframework.org/schema/security*  *http://www.springframework.org/schema/security/spring-security.xsd"*>  <sec:global-method-security pre-post-annotations=*"enabled"* secured-annotations=*"enabled"*/>  <!-- <sec:global-method-security secured-annotations="enabled"/> -->  <sec:http pattern=*"/css/\*\*"* security=*"none"* />  <sec:http pattern=*"/js/\*\*"* security=*"none"* />  <sec:http pattern=*"/images/\*\*"* security=*"none"* />  <sec:http pattern=*"/resources/\*\*"* security=*"none"* />  <sec:http auto-config=*"true"* use-expressions=*"true"*>  <!-- URL 접근제한을 주기위한 셋팅 DB에서불러와서 처리한다 -->  <sec:custom-filter ref=*"parancoeFilterSecurityInterceptor"* before=*"FILTER\_SECURITY\_INTERCEPTOR"* />  <!--  login-page 로그인 페이지 주소를 지정  username-parameter 로그인 페이지 form에 있는 username을 저장한 변수이름 지정  password-parameter 로그인 페이지 form에 있는 password을 저장한 변수이름 지정  login-processing-url 로그인 페이지 form action에 입력할 주소 지정  default-target-url 로그인 성공인 경우 호출할 주소 지정  authentication-failure-url 로그인 실패인 경우 호출할 주소 지정  always-use-default-target  -->  <sec:form-login login-page=*"/"*  username-parameter=*"username"*  password-parameter=*"password"*  login-processing-url=*"/login"*  default-target-url=*"/"*  authentication-failure-url=*"/"*  />  <sec:logout logout-url=*"/logout"* logout-success-url=*"/"* delete-cookies=*"JSESSIONID"*/>    <!--권한없는 페이지 호출시 web.xml쪽에서 따로 처리해준다-->  <sec:access-denied-handler error-page=*"/common/error/error403.jsp"*/>    <!-- 중복로그인 막아요 session cnt max 1-->  <sec:session-management>  <sec:concurrency-control max-sessions=*"1"* expired-url=*"/"*/>  </sec:session-management>  </sec:http>      <sec:authentication-manager>  <sec:authentication-provider ref=*"customAuthenticationProvider"*/>  </sec:authentication-manager>      <bean id=*"parancoeFilterSecurityInterceptor"* class=*"org.springframework.security.web.access.intercept.FilterSecurityInterceptor"* >  <property name=*"authenticationManager"* ref=*"org.springframework.security.authenticationManager"*/>  <property name=*"accessDecisionManager"* ref=*"accessDecisionManager"*/>  <property name=*"securityMetadataSource"* ref=*"customFilterInvocationSecurityMetadataSource"*/>  </bean>    <bean id=*"accessDecisionManager"* class=*"org.springframework.security.access.vote.AffirmativeBased"*>  <constructor-arg>  <list>  <bean class=*"org.springframework.security.access.vote.RoleVoter"*>  <property name=*"rolePrefix"* value=*""* />  </bean>  </list>  </constructor-arg>  <property name=*"allowIfAllAbstainDecisions"* value=*"false"* />  </bean>    <!-- 초기 부트됐을때 메뉴 리스트를 가져온걸로 페이지 호출시 매번 해당페이지가 관리포인트 인지 확인처리 합니다. -->  <bean id=*"customFilterInvocationSecurityMetadataSource"* class=*"com.ko.omnicns.omnifit.boot.security.CustomFilterInvocationSecurityMetadataSource"* />  <!-- 로그인 처리 -->  <bean id=*"customPasswordEncoder"* class=*"com.ko.omnicns.omnifit.login.security.CustomPasswordEncoder"* />  <bean id=*"customAuthenticationProvider"* class=*"com.ko.omnicns.omnifit.login.security.CustomAuthenticationProvider"*/>  </beans> |

다국어

|  |
| --- |
| <bean id=*"localeResolver"* class=*"org.springframework.web.servlet.i18n.SessionLocaleResolver"* />  <!-- 쿠키를 이용한 Locale 이용시 <bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver"/> -->  <bean id=*"localeChangeInterceptor"* class=*"org.springframework.web.servlet.i18n.LocaleChangeInterceptor"*>  <property name=*"paramName"* value=*"language"* />  </bean> |

Database source

|  |
| --- |
| <jdbc:embedded-database id=*"dataSource"* type=*"HSQL"*>  <jdbc:script location= *"classpath:/db/sampledb.sql"*/>  </jdbc:embedded-database>    <!-- JNDI -->  <bean id=*"dataSourceSpied"* class=*"org.springframework.jndi.JndiObjectFactoryBean"*>  <property name=*"jndiName"* value=*"jdbc/omnifitOracle"*/>  <property name=*"resourceRef"* value=*"true"*></property>  </bean>    <!-- JDBC LOG를 위한 설정 -->  <bean id=*"dataSource"* class=*"net.sf.log4jdbc.Log4jdbcProxyDataSource"*>  <constructor-arg ref=*"dataSourceSpied"* />  <property name=*"logFormatter"*>  <bean class=*"net.sf.log4jdbc.tools.Log4JdbcCustomFormatter"*>  <property name=*"loggingType"* value=*"MULTI\_LINE"* />  <property name=*"sqlPrefix"* value=*"SQL:::"*/>  </bean>  </property>  </bean> |

viewResolver

|  |
| --- |
| <!-- view Resolver Setting tiles3  <bean id="tilesConfigurer" class="org.springframework.web.servlet.view.tiles3.TilesConfigurer">  <property name="definitions">  <value>/WEB-INF/config/tiles/tiles-layouts.xml</value>  </property>  </bean>  <bean class="org.springframework.web.servlet.view.UrlBasedViewResolver" p:order="1"  p:viewClass="org.springframework.web.servlet.view.tiles3.TilesView"/>  -->      <!-- view Resolver Setting UrlBasedViewResolver-->  <bean class=*"org.springframework.web.servlet.view.UrlBasedViewResolver"* p:order=*"2"*  p:viewClass=*"org.springframework.web.servlet.view.JstlView"*  p:prefix=*"/WEB-INF/jsp/"* p:suffix=*".jsp"*/> |

Aop

|  |
| --- |
| <!-- aop 사용하지않음  <bean id="PrivacyAOP" class="com.ko.omnicns.omnifit.security.encrypt.PrivacyAOP"/>  <bean id="UserService" class="com.ko.omnicns.omnifit.cms.user.service.UserServiceImpl"/>  <aop:aspectj-autoproxy />  --> |
| @Aspect  **public** **class** ~~PrivacyAOP~~ {  // @Pointcut("execution(public \* com.ko.omnicns.omnifit.cms.user.service..\*Impl.\*(..))")  **public** PrivacyAOP() {  System.***out***.println("--PrivacyAOP-");  }  //com.ko.omnicns.omnifit.cms.sample  // com.ko.omnicns.omnifit.cms.sample.service.CMSSampleServiceImpl  // @Before("execution(\* com.ko.omnicns.omnifit.cms.sample.service..\*Impl..\*(..))")  // @Before("execution(public \* com.ko.omnicns.omnifit.cms.sample.service..\*Impl..\*(..))")  // @Before("execution(public \* com.ko.omnicns.omnifit.cms.user.service.UserMapper..\*(..))")  // com.ko.omnicns.omnifit.cms.user.service.selectUserPermission(..)  // @Before("execution(public \* com.ko.omnicns.omnifit.cms..\*.service..\*..\*(..))")  // @Before("execution(public \* com.ko.omnicns.omnifit.cms..\*.service..\*Impl..\*(..))")      // @Before("execution(\* com.ko.omnicns.omnifit.cms.user.service.UserServiceImpl..\*(..))")    // @Before("execution(\* com.ko.omnicns.omnifit.cms.sample.CMSSampleController..\*(..))")  // @Before("execution(\* com.ko.omnicns.omnifit.cms.user.UserController..\*(..))")  //execution(\* com.ko.omnicns.omnifit.\*\*.\*Impl.\*(..))  //com.ko.omnicns.omnifit.cms.\*\*.service..\*Impl..\*(..)  // @Before("execution(com.ko.omnicns.omnifit.cms.\*\*.service..\*Impl.\*(..)")  // @Before("execution(public \* com.ko.omnicns.omnifit.cms.\*\*.service..\*Impl.\*(..))")  **public** **void** trace(JoinPoint joinPoint) **throws** Throwable {  System.***out***.println("--PrivacyAOP-"+joinPoint);  }  } |

Global Exception Handler

|  |
| --- |
| <!-- global Exception -->  <bean class=*"c*  @ControllerAdvice("com.ko.omnicns.omnifit.cms")  **public** **class** CustomException {  @ResponseStatus(value=HttpStatus.***NOT\_FOUND***, reason="IOException occured")  @ExceptionHandler(RuntimeException.**class**)  **public** String handlerRuntimeException(){  **return** "cmmn/exception";  }  }  *om.ko.omnicns.omnifit.comm.exception.CustomException"* /> |
| http://www.journaldev.com/2651/spring-mvc-exception-handling-exceptionhandler-controlleradvice-handlerexceptionresolver-json-response-example |
| @ControllerAdvice("com.ko.omnicns.omnifit.cms")  **public** **class** CustomException {      @ExceptionHandler(ErrosException.**class**)  @ResponseBody  **public** String errosexception(ErrosException errors){  **return** **new** JsonVO(errors.getErrors()).toJson();  }      @ExceptionHandler(Exception.**class**)  @ResponseBody  **public** ResponseEntity<?> handleAnyException(Exception e) {  **return** errorResponse(e, HttpStatus.***INTERNAL\_SERVER\_ERROR***);  }  **protected** ResponseEntity<String> errorResponse(  Throwable throwable, HttpStatus status) {  **if** (**null** != throwable) {  **return** response(**new** JsonVO(JsonHead.***CODE\_ERROR***,throwable.getMessage()).toJson(), status);  } **else** {  **return** response(**null**, status);  }  }  **protected** <T> ResponseEntity<T> response(T body, HttpStatus status) {  **return** **new** ResponseEntity<T>(body, **new** HttpHeaders(), status);  }  } |
| Controller  @RequestMapping(value="/indexTest2", method = {RequestMethod.***GET***, RequestMethod.***POST***},produces = "application/json; charset=utf8")  @ResponseBody  **public** String indexTest2(@Valid @ModelAttribute("FromToVO") FromToVO ft,Errors erros, HttpServletRequest request, ModelMap model) **throws** Exception {  log.debug(ft.getFrom());  log.debug(ft.getTo());  **if**(erros.hasErrors()){  **throw** **new** ErrosException(erros);  }  log.debug("1");  **int** x = 0;  log.debug("2"+x);  **int** y = 1;  log.debug("3"+y);  **int** z = y / x;  log.debug("4"+z);    **return** "{}";  } |

DI

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:c=*"http://www.springframework.org/schema/c"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xsi:schemaLocation=*"http://www.springframework.org/schema/beans*  *http://www.springframework.org/schema/beans/spring-beans.xsd"*>  <bean id=*"user1"* class=*"net.madvirus.spring4.chap02.User"*  c:id=*"bkchoi"* c:password=*"1234"* />    <bean id=*"user2"* class=*"net.madvirus.spring4.chap02.User"*  c:\_0=*"madvirus"* c:\_1=*"qwer"* />  <bean id=*"pwChangeSvc"*  class=*"net.madvirus.spring4.chap02.PasswordChangeService"*  c:userRepository-ref=*"userRepository"* />    <bean id=*"authFailLogger"* class=*"net.madvirus.spring4.chap02.AuthFailLogger"*  p:threshold=*"2"* />  <bean id=*"authenticationService"*  class=*"net.madvirus.spring4.chap02.AuthenticationService"*  p:failLogger-ref=*"authFailLogger"*  p:userRepository-ref=*"userRepository"* />    <bean id=*"userRepository"* class=*"net.madvirus.spring4.chap02.UserRepository"*>  <property name=*"users"*>  <list>  <ref bean=*"user1"* />  <ref bean=*"user2"* />  </list>  </property>  </bean>  </beans> |