

경량환경 템플릿이란?

홈페이지, 업무시스템, 포털사이트와 같은 웹시스템의 개발시의 활용 가능한 참조용 프로젝트 소스이며 공통컴포넌트의 기능들 중 일부를 선정하여 기본 구성하였다. Eclipse 개발 환경에서 제공하는 Wizard 기능을 이용하여 프로젝트를 간편하게 생성할 수 있으며 포함된 DB script를 사용하여 필요 테이블을 생성한 후 Tomcat과 같은 서버릿 엔진을 사용하여 테스트 구동해 볼 수 있다.

- 경량환경 템플릿 소개(Introduction)
- 경량환경 템플릿이란?
- 경량환경 템플릿의 기능 구성
- 홈페이지 템플릿 메뉴구성
- 내부업무 템플릿 메뉴구성
- 포털사이트 템플릿 메뉴구성

경량환경 템플릿의 기능 구성

경량환경 템플릿은 각 테이블별로 공통컴포넌트에서 기본기능을 선정하여 기본기능을 구성하였으며 전체 시스템의 모습을 갖추기 위해 예시형태의 메뉴가 포함된 메뉴구조로 구성되었다.(예시메뉴에 해당하는 기능은 html형태의 화면뷰만 제공됨) 홈페이지 템플릿은 게시판, 일정관리(부서일정)를 기본기능으로 차용한후 프로젝트 형태로 구성되었다. 내부업무 템플릿은 내부적인 시스템관리 기능을 위주로 차용하여 구성되었고, 포털사이트 템플릿은 설문, FAQ 등의 커뮤니케이션 관련 기능을 위주로 차용하여 구성되었다. 경량환경 템플릿 소스는 기본 구성에 사용한 공통컴포넌트 소스가 커스터마이징되어 각 템플릿 프로젝트에 맞게 변경되었다. 만일 공통컴포넌트의 오리지널 버전형태의 기능을 사용하려면 개발환경에서 제공하는 공통컴포넌트 소스를 조립하는 Wizard를 사용하여 재구성한다.

템플릿 종류	기본기능(기능차용 및 커스터마이징)	비고
홈페이지	게시판관리, 일정관리(부서일정)	메뉴, 관리자로그인 추가구성
내부업무	사용자관리, 부재관리, 권한관리, 프로그램관리, 메뉴관리, 코드관리, 우편번호관리, 로그관리, 통계관리)	메뉴 추가구성
포털 사이트	회원관리, 약관관리, 개인정보보호정책관리, 권한관리, 설문관리, 배너관리, FAQ, Q&A, 게시판관리	메뉴 추가구성

홈페이지 템플릿 메뉴구성

대메뉴	서브메뉴	비고
사이트 소개	사이트소개, 연혁, 조직소개, 찾아오시는길	예시메뉴(템플릿 샘플 htm)
정보마당	주요사업소개, 대표서비스소개	예시메뉴(템플릿 샘플 htm)
고객지원	자료실, 묻고답하기	예시메뉴(템플릿 샘플 htm)
알림마당	오늘의행사, 금주의행사, 공지사항, 사이트갤러리	
사이트 관리	일정관리, 게시판템플릿관리, 게시판생성관리, 게시물관리(공지사항),게시물관리(갤러리)	

[홈페이지 템플릿 메뉴구성]

사이트소개	정보마당	고객지원	알림마당	사이트관리
사이트소개 연혁 조직소개 찾아오시는길	주요사업소개 대표서비스소개	자료실 묻고답하기	오늘의행사 금주의행사 공지사항 사이트갤러리	일정관리 게시판템플릿관리 게시판생성관리 게시물관리 공지사항관리 사이트갤러리관리

외부사용자용 메뉴

관리자용 메뉴

내부업무 템플릿 메뉴구성

대메뉴	서브메뉴	비고
알림 정보	공지사항, 업무게시판	
직급체 계관리	입퇴사정보관리, 직급정보관리, 직위정보관리	예시메뉴(템플릿 샘플 html 제공)
진급 관리	업무평가점수 관리, 상벌정보 관리	예시메뉴(템플릿 샘플 html 제공)
근태 관리	출퇴근정보 관리, 휴무정보 관리	예시메뉴(템플릿 샘플 html 제공)
내부 서비스 관리	게시판관리(게시판템플릿관리, 게시판생성관리, 게시판사용여부관리, 공지사항관리, 업무게시판관리), 사용현황관리 (접속로그관리, 접속통계관리, 로그인정책관리)	
내부 시스템 관리	사용자관리(사용자등록관리, 사용자부재관리), 사용자권한관리(권한관리, 그룹관리, 사용자별권한관리, 롤관리), 메뉴관리(프로그램관리, 메뉴관리, 메뉴생성관리), 코드관리(분류코드관리, 공통코드관리, 상세코드관리)	

알림정보	직급관리	진급관리	근태관리	서비스관리	시스템관리
공지사항 업무게시판	입퇴사관리 직급관리 직위관리	업무평가관리 상별정보관리	출퇴근관리 휴무정보관리	게시판관리 게시판템플릿관리 게시판생성관리 게시판사용관리 공지사항관리 업무게시판관리 사용현황관리 접속로그관리 접속통계관리 로그인정책관리	사용자관리 등록관리 부재관리 사용자권한관리 권한관리 그룹관리 사용자별권한 롤관리 메뉴관리 프로그램관리 메뉴관리 메뉴생성관리 코드관리 분류코드관리 공통코드관리 상세코드관리

업무사용자용 메뉴

관리자용 메뉴

포털사이트 템플릿 메뉴구성

대메뉴	서브메뉴	비고
사이트 소개	사이트소개, 연혁, 조직소개, 찾아오시는길	예시메뉴(템플릿 샘플 html 제공)
민원 광장	민원신청, 민원결과확인	예시메뉴(템플릿 샘플 htm)
알림 마당	공지사항, 자유게시판	
정보 마당	FAQ, Q&A, 설문조사	
포털 서비스 관리	설문관리(설문지, 설문템플릿, 설문문항, 설문항목, 설문응답결과), 서비스관리(FAQ관리, QA관리, 게시판템플릿관리, 계산생성관리, 게시판사용관리, 공지사항관리, 자유게시판관리, 배너관리)	
포털 시스템 관리	사용자관리(회원관리, 이용약관관리, 개인정보보호관리), 사용자권한관리(권한관리, 그룹관리, 사용자별권한관리, 롤관리)	

사이트소개	민원광장	알림마당	정보마당	서비스관리	시스템관리
사이트소개 연혁 조직소개 찾아오는길	민원신청 민원결과확인	공지사항 자유게시판	FAQ Q&A 설문조사	설문관리 설문지 설문템플릿 설문문항 설문항목 설문응답결과 서비스관리 FAQ관리 QA관리 게시판템플릿관리 게시판생성관리 게시판사용관리 공지사항관리 자유게시판관리	사용자관리 회원관리 이용약관관리 개인정보정책관리 사용자권한관리 권한관리 그룹관리 사용자별권한 롤관리

일반회원용 메뉴

관리자용 메뉴


경량환경 템플릿 다운로드

- 경량환경 템플릿 사용하기(use)
- 경량환경 템플릿 다운로드
- 개발환경도구의 템플릿 사용 절차

템플릿 소스는 개발자용 개발환경도구(전자정부 프레임워크용 플러그인이 탑재된 eclipse)를 내려받게 되면 홈페이지, 업무 시스템, 포털사이트에 대한 템플릿 소스를 내려받은 상태라고 볼 수 있다.
각 템플릿의 소스는 개발도구의 템플릿 프로젝트 생성 메뉴를 통해서 손쉽게 확인할 수 있으면 테스트용 테이블을 생성한 후 (프로젝트 소스에 포함되어 있는 DATABASE 폴더의 테이블 생성스크립트를 사용) 곧바로 실행해 볼수 있다.
템플릿 메뉴 생성 기능은 개발환경 도구의 매뉴얼을 참조한다.

개발환경도구의 템플릿 사용 절차

1. 개발환경 설치

프레임워크 사이트에서 개발자용 개발환경 도구를 다운로드 후 설치한다. ([개발환경 도구의 다운로드와 설치 방법 바로가기](#))
⇒ 테스트 DB환경이 없는 경우 HSQL DB 내려받기(압축해제 후 runHsqlDB.cmd 실행하면 홈페이지 템플릿 테스트 가능함)
:  [다운로드](#)

2. 템플릿 프로젝트 생성(템플릿 생성 메뉴 사용)

개발환경 도구의 템플릿 생성 메뉴를 통해서 개발환경 템플릿 프로젝트를 생성한다. ([템플릿 프로젝트 메뉴 사용방법 바로가기](#))

3. 개발환경 프로젝트 실행

템플릿 프로젝트의 메뉴구성 및 기능 설명을 확인한다. ([경량환경 템플릿 기본기능 설명 바로가기](#))

4. 개발환경 프로젝트 환경설정

템플릿 프로젝트의 메뉴구성 및 기능 설명을 확인한다. ([경량환경 템플릿 환경설정 바로가기](#))

5. 템플릿의 기본기능관련 참조사항


- 홈페이지 템플릿의 관리자 암호 변경

홈페이지 템플릿에는 사용자 정보 변경기능이 포함되어 있지 않으므로 변경이 필요한 경우는 DB에서 직접 수정해야 합니다.
암호변경을 하고자 하는 경우는 암호변경용 JSP 페이지를 다운로드하여 사용할 수 있습니다(index.jsp와 동일한 위치에 저장한후 실행)

(암호변경 JSP  [다운로드](#))

- 포털사이트 템플릿의 메뉴구조 변경

포털사이트 템플릿에는 메뉴, 프로그램 변경기능이 포함되어 있지 않으므로 변경이 필요한 경우는 DB에서 직접 수정해야 합니다.
포털사이트 템플릿의 메뉴구조는 프로그램목록정보(LETTNPROGRMLIST), 메뉴 (LETTNMENUINFO)를 통해 확인할 수 있다.
로그인시 보이는 메뉴는 메뉴생성정보(LETTNMENUCREATDTLS)에 있는 권한별 메뉴목록 정보를 통해 실제 구성한다.
첨부된 엑셀파일을 참조하여 필요시 편집작업을 통해 전체적으로 재구성하여 DB에 반영할 수 있다.

(프로그램,메뉴,메뉴생성정보 엑셀파일  [다운로드](#))

경량환경 템플릿 구성 및 환경설정(configuration)

Table of Contents

- 경량환경 템플릿 구성 및 환경설정 (configuration)
 - web.xml 주요 설정
 - egovframework/spring/com/context-*.xml 주요 설정
 - /WEB-INF/config/egovframework/springmvc/*.xml 주요 설정
 - datasource관련 설정

- 템플릿 소스의 웹환경 관련 설정은 아래 3곳의 위치에 정의되어 사용된다.
 - web.xml
 - classpath*:egovframework/spring/com/context-*.xml
 - /WEB-INF/config/egovframework/springmvc/*.xml
- 템플릿 소스의 DB관련 설정은 globlas.properties에 정의되어 사용된다.

web.xml 주요 설정

- *.do로 호출되는 경우 기본 화면에 출력되는 encoding은 utf-8로 필터링 되도록 설정하여 특수문자에 의한 크로스사이트 스크립트를 방지한다. HTMLTagFilter를 설정하면 파라미터로 전달되는 값 자체를 변환하는 방식의 사용도 가능하다. 템플릿 소스에서는 화면을 통해 입력하는 값을 변환하지 않고 출력하는 시점에 encoding처리하는 방식을 기본으로 사용한다.

```
<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>
    org.springframework.web.filter.CharacterEncodingFilter
  </filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>utf-8</param-value>
  </init-param>
</filter>
<filter>
  <filter-name>HTMLTagFilter</filter-name>
  <filter-class>
    egovframework.rte.ptl.mvc.filter.HTMLTagFilter
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
<!-- 템플릿은 UTF-8 인코딩 필터와 JSP 출력시 <c:out> 태그 사용이 기본정책 -->
<!-- 아래의 HtmlTagFilter를 추가등록하고 기능을 보강하여 사용할 수도 있음 -->
<!--
<filter-mapping>
  <filter-name>HTMLTagFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
-->
```

- spring context관련 환경설정정보 파일 위치를 등록한다.

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    classpath*:egovframework/spring/com/context-*.xml
  </param-value>
</context-param>
```

- spring mvc를 위한 Listener, DispatcherServlet설정과 mvc관련 관련 추가 환경설정정보 파일위치를 등록한다.

```
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>
    org.springframework.web.servlet.DispatcherServlet
  </servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
      /WEB-INF/config/egovframework/springmvc/*.xml
    </param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

- spring security 를 사용하는 경우 포함되는 설정(내부업무, 포털사이트 템플릿만 포함됨)

```
<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<listener>
  <listener-class>org.springframework.security.ui.session.HttpSessionEventPublisher</listener-class>
</listener>
```

egovframework/spring/com/context-*.xml 주요 설정

추가 설정파일들은 템플릿 프로젝트 소스의 각 resource/egovframework/spring/com/ 폴더 아래에 위치한다. anotation, aop, schedule, validation, security, id-generation, sqlmap, transaction 관련 설정정보가 등록된다. (템플릿 종류별로 차이 있음)

- context-aspect.xml의 ExceptionTransfer 설정(템플릿 서비스임플 클래스에서 발생하는 경우는 Exception에 대한 개별 로그 출력)

```
<aop:config>
  <aop:pointcut id="serviceMethod"
    expression="execution(* egovframework.let..impl.*Impl.*(..)) or execution(* egovframework.com..impl.*Impl.*(..))"
  />

  <aop:aspect ref="exceptionTransfer">
    <aop:after-throwing throwing="exception"
      pointcut-ref="serviceMethod" method="transfer" />
    </aop:aspect>
  </aop:config>
  ...
```

- context-common.xml의 egovMessageSource 설정(프로퍼티 파일에 등록된 메시지 활용), leaveTrace 설정(Exception발생시 개별 Trace 출력)

```
<bean id="egovMessageSource" class="egovframework.com.cmm.EgovMessageSource">
  <property name="reloadableResourceBundleMessageSource">
    <ref bean="messageSource" />
  </property>
</bean>
...

<!-- Exception 발생시 후처리용 별도작업을 위해 실행환경의 LeaveTrace를 활용하도록 설정 -->
<bean id="leaveTrace" class="egovframework.rte.fdl.cmmn.trace.LeaveTrace">
  <property name="traceHandlerServices">
    <list>
      <ref bean="traceHandlerService" />
    </list>
  </property>
</bean>
...

<!-- 소스내에서 egovMessageSource의 사용 예
@Resource(name="egovMessageSource")
EgovMessageSource egovMessageSource;
String message = egovMessageSource.getMessage("fail.common.login");
-->

<!-- 소스내에서 사용 예(템플릿에서는 특별한 후속처리 없으므로 로그아웃시에 샘플소스만 적용함)
@Resource(name="leaveTrace") LeaveTrace leaveTrace;
...
} catch (ArithmeticException athex) {
  // TraceHandler에서 메시지에 따른 후속조치 등을 처리할 수 있다.
  leaveTrace.trace("fail.common.msg", this.getClass());
}
...
-->
```

- context-datasource.xml의 datasource 설정

```
<!-- globals.properties 파일에 등록된 DB정보를 사용하여 datasource 정보를 정의한다. -->
<bean id="propertyConfigurer"
  class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
  <property name="locations">
    <list>
      <value>classpath:/egovframework/egovProps/globals.properties</value>
    </list>
  </property>
</bean>

<!-- datasource 설정(propertyConfigurer 활용) -->
<alias name="dataSource-${Globals.DbType}" alias="dataSource" />
<alias name="dataSource-${Globals.DbType}" alias="egov.dataSource" />

<!-- mysql -->
<bean id="dataSource-mysql" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
  <property name="driverClassName" value="${Globals.DriverClassName}" />
  <property name="url" value="${Globals.Url}" />
  <property name="username" value="${Globals.UserName}" />
  <property name="password" value="${Globals.Password}" />
</bean>
...

<!-- DB 4개 종류에 대한 정의의 샘플이 존재한다. -->
<!-- 홈페이지 템플릿 globals.properties의 dbtype을 hsql로 변경후 아래 설정을 사용하여 간단한 테스트가 가능하다. -->
<bean id="dataSource-hsql" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
  <property name="driverClassName" value="net.sf.log4jdbc.DriverSpy" />
  <property name="url" value="jdbc:log4jdbc:hsqldb:hsqldb://localhost/sampled" />
  <property name="username" value="sa" />
</bean>
```

- context-idgen.xml의 업무별 일련번호 생성 서비스 설정

```
<!-- 시스템에서 업무별로 레코드 신규발생시 일련번호를 생성하는 bean 클래스 설정 -->
<!-- 첨부파일 ID Generation Config-->
<bean name="egovFileIdGnrService"
  class="egovframework.rte.fdl.idgnr.impl.EgovTableIdGnrService"
  destroy-method="destroy">
  <property name="dataSource" ref="dataSource-${Globals.DbType}" />
  <property name="strategy" ref="fileStrategy" />
  <property name="blockSize" value="10" />
  <property name="table" value="IDS" />
  <property name="tableName" value="FILE_ID" />
</bean>

<!-- 첨부파일 ID Generation Strategy Config -->
<bean name="fileStrategy"
  class="egovframework.rte.fdl.idgnr.impl.strategy.EgovIdGnrStrategyImpl">
  <property name="prefix" value="FILE_" />
  <property name="ciphers" value="15" />
  <property name="fillChar" value="0" />
</bean>
...

<!-- 소스내 사용 예
...
@Resource(name = "egovFileIdGnrService") private EgovIdGnrService idgenService;
```



```
...
String newId = idgenService.getNextStringId();
...
-->
```

■ context-properties.xml의 프로퍼티 값 설정

```
<!-- 프로퍼티 정보 설정 -->
<bean name="propertiesService" class="egovframework.rte.fdl.property.impl.EgovPropertyServiceImpl" destroy-
method="destroy">
  <property name="properties">
    <map>
      <entry key="pageUnit" value="10"/>
      <entry key="pageSize" value="10"/>
      <entry key="posblAtchFileSize" value="5242880"/>
      <entry key="Globals.fileStorePath" value="/user/file/sht/" />
      <entry key="Globals.addedOptions" value="false" />
    </map>
  </property>
</bean>

<!-- 소스 내 사용 예
@Resource(name = "propertiesService") protected EgovPropertyService pService;
String savePath = pService.getString("Globals.fileStorePath");
...
-->
```

■ context-sqlMap.xml의 프로퍼티 값 설정

```
<!-- 소스내 DAO에서 사용할 쿼리문의 위치정보를 확인 bean 설정으로 Dao 클래스에서 extends하는 AbstractDao내부에 포함하여 사용 -->
<!-- configLocations properties 항목에 쿼리문 위치정보 파일들을 등록( 각 위치정보 파일내에 실제 쿼리문들이 포함된 파일들의 위치정보가 존
재) -->
<!-- SqlMap setup for iBATIS Database Layer -->
<bean id="sqlMapClient" class="org.springframework.orm.ibatis.SqlMapClientFactoryBean">
  <property name="configLocations">
    <list>
      <value>classpath:/egovframework/sqlmap/config/${Globals.DbType}/*.xml</value>
    </list>
  </property>
  <property name="dataSource" ref="dataSource-${Globals.DbType}" />
  <property name="lobHandler" ref="lobHandler" />
</bean>
...

<!-- 소스내 사용예 (DAO 클래스 정의시 extends를 통해 자동으로 사용 -->
<!-- com.sqlMapClient 와 같이 새로 지정한후 EgovAbstractDAO를 재정의하여 사용할수도 있음(소스내 설정 참고) -->
@Repository("BBSAttributeManageDAO")
public class BBSAttributeManageDAO extends EgovAbstractDAO {
  ...
-->
```

* context-transaction.xml의 트랜잭션 설정

```
<bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource"/>
</bean>

<tx:advice id="txAdvice" transaction-manager="txManager">
  <tx:attributes>
    <tx:method name="*" rollback-for="Exception"/>
  </tx:attributes>
</tx:advice>

<aop:config>
  <aop:pointcut id="requiredTx"
    expression="execution(* egovframework.let..impl.*Impl.*(..)) or execution(*
egovframework.com..*Impl.*(..))"/>
  <aop:advisor advice-ref="txAdvice"
    pointcut-ref="requiredTx" />
</aop:config>

<!-- 템플릿에서는 서비스 임플 클래스내의 함수내에서 트랜잭션을 처리한다.(Exception발생시 임플클래스의 함수내에서 호출된 DAO 함수들의 처
리결과가 모두 롤백됨) -->
```

* context-validator.xml의 Validator설정

```
<bean id="beanValidator" class="org.springframework.validation.commons.DefaultBeanValidator">
  <property name="validatorFactory" ref="validatorFactory"/>
</bean>

<bean id="validatorFactory" class="org.springframework.validation.commons.DefaultValidatorFactory">
  <property name="validationConfigLocations">
    <list>
      <!-- 경량환경 템플릿 밸리데이터 설정 -->
      <value>classpath:/egovframework/validator/validator-rules-let.xml</value>
      <value>classpath:/egovframework/validator/let/**/*.xml</value>
    </list>
  </property>
</bean>

<!-- validationConfigLocations 프로퍼티로 등록된 xml파일내에 validation시 사용될 함수와 validation의 대상이 되는 form 정보가 등록된
다. -->
<!-- classpath:/egovframework/validator/validator-rules-let.xml 내용 예
...
<validator name="required"
  <javascript><![CDATA[
    function validateRequired(form) {
      var isValid = true;
      ...
    }
  -->
<!-- classpath:/egovframework/validator/let/*.xml 내용 예
...
<validator name="required"
  <javascript><![CDATA[
    function validateRequired(form) {
      var isValid = true;
      ...
    }
  -->
<!--
```



```

<formset>
    <form name="boardMaster">
        <field property="bbsNm" depends="required">
            <arg0 key="cop.bbsNm" />
        </field>
        ...
    </form>
-->

```

* context-security.xml의 스프링 시큐리티 설정 (내부업무 템플릿, 포탈사이트 템플릿 해당)

```

<!-- 스프링시큐리티 관련 bean들에 대한 설정으로 jdbcService를 사용한 권한컨트롤 설정을 조정하여 사용할 수 있다. -->
<b:bean id="jdbcUserService"
    class="egovframework.let.sec.security.userdetails.jdbc.EgovJdbcUserDetailsManager" >
    <b:property name="usersByUsernameQuery" value="SELECT USER_ID, ESNTL_ID AS PASSWORD, 1 ENABLED, USER_NM, USER_ZIP,
    USER_ADRES, USER_EMAIL, USER_SE, usr.ORGNZT_ID, ESNTL_ID, org.ORGNZT_NM FROM COMVNUSERMASTER usr, LETTNORGNZTINFO org WHERE
    usr.ORGNZT_ID = org.ORGNZT_ID AND CONCAT(USER_SE, USER_ID) = ? " />

    <b:property name="authoritiesByUsernameQuery" value="SELECT A.SCRTY_DTRMN_TRGET_ID USER_ID, A.AUTHOR_CODE AUTHORITY
    FROM LETTNEMPLYRSCRTYESTBS A, COMVNUSERMASTER B WHERE A.SCRTY_DTRMN_TRGET_ID = B.ESNTL_ID AND B.USER_ID = ? " />
    <b:property name="roleHierarchy" ref="roleHierarchy" />
    <b:property name="dataSource" ref="dataSource" />
    <b:property name="mapClass" value="egovframework.let.sec.security.common.EgovSessionMapping" />
</b:bean>

```

* context-excel.xml의 excel파일 컨트롤 설정(내부업무 템플릿 해당)

```

<bean id="excelZipService" class="egovframework.rte.fdl.excel.impl.EgovExcelServiceImpl" >
    <property name="propertyPath" value="excelInfo.xml" />
    <property name="mapClass" value="egovframework.let.sym.ccm.zip.service.impl.EgovCcmExcelZipMapping" />
    <property name="sqlMapClient" ref="sqlMapClient" />
</bean>
<!-- 소스내 사용 예 (엑셀컬럼 정보를 설정한후 정보 설정한 후 (템플릿의 EgovCcmExcelZipMapping 클래스 참조) 엑셀업로드시 uploadExcel 함
수 사용)
    excelZipService.uploadExcel("ZipManageDAO.insertExcelZip", file, 2, (long) 5000);
-->

```

* context-scheduling.xml의 일단위 접속통계 스케줄링 설정(내부업무 템플릿 해당)

```

<!-- 시스템 로그 요약 -->
<bean id="sysLogging"
    class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean" >
    <property name="targetObject" ref="egovSysLogScheduling" />
    <property name="targetMethod" value="sysLogSummary" />
    <property name="concurrent" value="false" />
</bean>
<bean id="sysLogTrigger" class="org.springframework.scheduling.quartz.SimpleTriggerBean" >
    <property name="jobDetail" ref="sysLogging" />
    <!-- 시작하고 1분후에 실행한다. (milisecond) 60000 -->
    <property name="startDelay" value="30000" />
    <!-- 매 24시간마다 실행한다.864000000 (milisecond) -->
    <property name="repeatInterval" value="864000000" />
</bean>
<bean id="sysLogScheduler"
    class="org.springframework.scheduling.quartz.SchedulerFactoryBean" >
    <property name="triggers">
        <list><ref bean="sysLogTrigger" /></list>
    </property>
</bean>

<!-- 서버가동후 1분후 시점을 기준으로 일단위로 egovSysLogScheduling서비스의 sysLogSummary()를 실행 -->

```

* context-syslogaop.xml의 aop를 사용한 로그 기록 설정(내부업무 템플릿 해당)

```

<bean id="logManage" class="egovframework.let.sym.log.clg.service.EgovLoginLogAspect" />
<aop:config>
    <aop:aspect id="logManageAspect" ref="logManage">
        <!-- 로그인 Method -->
        <aop:before pointcut="execution(public * egovframework.let.uat.uia.service.EgovLoginService.actionLogin(...))"
        method="logLogin" />
    </aop:aspect>
</aop:config>

<!-- 로그인 관련 서비스 클래스내에서 actionLogin으로 시작하는 함수 동작시 logManage를 통해 접속로그 등록 -->

```

/WEB-INF/config/egovframework/springmvc/*.xml

추가설정파일들은 템플릿 프로젝트 소스의 각 WEB-INF/config/egovframework/springmvc 폴더 아래에 위치한다.
exception, view, error-page 관련 설정 정보가 등록된다.

■ auto detect를 위한 mvc 설정

```

<context:component-scan base-package="egovframework">
    <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller" />
    <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Service" />
    <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Repository" />
</context:component-scan>

```

■ anotation request mapping과 commandmap형태의 argument를 전달을 위한 설정

```

<bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter">
    <property name="webBindingInitializer">
        <bean class="egovframework.com.cmm.web.EgovBindingInitializer" />
    </property>
    <property name="customArgumentResolvers">
        <list>
            <bean class="egovframework.rte.ptl.mvc.bind.CommandMapArgumentResolver" />
        </list>
    </property>
</bean>
<bean class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerMapping" />

```

- exception 발생시 이동할 오류페이지 설정

```
<bean class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">
  <property name="defaultErrorView" value="cmm/error/egovError"/>
  <property name="exceptionMappings">
    <props>
      <prop key="org.springframework.dao.DataAccessException">cmm/error/dataAccessFailure</prop>
      <prop key="org.springframework.transaction.TransactionException">cmm/error/transactionFailure</prop>
      <prop key="egovframework.rte.fdl.cmm.exception.EgovBizException">cmm/error/egovError</prop>
      <prop key="org.springframework.security.AccessDeniedException">cmm/error/accessDenied</prop>
    </props>
  </property>
</bean>
```

- 화면처리용 JSP 파일명의 prefix, suffix 처리에 대한 설정

```
<bean class="org.springframework.web.servlet.view.UrlBasedViewResolver" p:order="1"
  p:viewClass="org.springframework.web.servlet.view.JstlView"
  p:prefix="/WEB-INF/jsp/" p:suffix=".jsp"/>
```

- 로그인에 필요한 URL을 체크하는 필터설정(홈페이지 템플릿만 사용하고 다른 템플릿은 시큐리티 권한 적용)

```
<!--홈페이지 템플릿은 로그인한 관리자에게 허용되는 URL에 대한 제약이 필터설정과 함께 Controller클래스내에서 접근체크함수를 통해서도 체크한다.-->
<!--필터설정이 해제되어도 소스내에서 체크로직이 존재하므로 제약 변경시 해당 Request Mapping(**.do)에 해당하는 Controller내의 소스 수정요함 -->
<bean id="selectAnnotationMapper"
  class="egovframework.rte.ptl.mvc.handler.SimpleUrlAnnotationHandlerMapping"
  p:order="1">
  <property name="interceptors">
    <list>
      <ref local="authenticInterceptor" />
    </list>
  </property>
  <property name="urls">
    <set>
      <!-- 게시판 생성, 템플릿 생성은 로그인후 사용 가능하다. 관리자만 사용가능 -->
      <value>/cop/com/*.do</value>
      <value>/cop/bbs/*Master*.do</value>
      <value>/uat/uia/*.do</value>
    </set>
  </property>
</bean>

<bean id="authenticInterceptor" class="egovframework.com.cmm.interceptor.AuthenticInterceptor" >
  <!-- 로그인 체크 인터셉터의 접근허용 URL 리스트-->
  <property name="permittedURL">
    <set>
      <value>/uat/uia/actionLogin.do</value>
      <value>/uat/uia/egovLoginUsr.do</value>
    </set>
  </property>
</bean>
```

datasource관련 설정

- globals.properties 주요 설정정보(datasource.xml의 설정 정보로 적용)

```
database 종류, database dirver, database 계정/암호 등을 설정한다.
템플릿 3종 모두 동일하게 적용되는데 홈페이지의 경우는 Hsql DB를 사용할 수 있는 설정이 추가로 포함되어 있다.
# DB서버 타입(mysql,oracle,altibase,tibero) - datasource 및 sqlMap 파일 지정에 사용됨
Globals.DbType = mysql
# DB 사용자 계정
Globals.UserName=sht
# DB 사용자 암호
Globals.Password=sht01
# driver 정보
Globals.DriverClassName=net.sf.log4jdbc.DriverSpy
# DB 연결 정보
Globals.Url=jdbc:mysql://192.168.200.24:1623/SHT
```

홈페이지 템플릿

- 경량환경 템플릿 기능(function)
 - 홈페이지 템플릿
 - 내부업무 템플릿
 - 포털사이트 템플릿

■ 홈페이지 템플릿관련 기본사항

- 1. 홈페이지 템플릿은 관리자만 로그인가능하다는 전제하여 구성되었으며 최초 관리자계정 설정은 [로그인계정 : admin , 로그인암호 : 1] 로 설정되어 있다.
- 2. 관리자 추가 및 변경기능은 추가구성되어 있지 않으므로 필요시 DB를 통해 직접 변경한다.(암호셋팅값은 공통컴포넌트의 암호화 로직에 따라 생성해야 함)
- 3. 기본기능이나 예시메뉴를 실무적으로 추가 커스터마이징하여 활용할 수 있다.

■ 사이트 소개

- 1. 세부메뉴 : 사이트소개, 연혁, 조직소개, 찾아오시는 길
- 2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 링크와 JSP파일이 존재한다.
- 3. 활용방법 : 각 샘플페이지에 대한 콘텐츠를 새로 구성하여 활용할 수 있다.

■ 정보마당

- 1. 세부메뉴 : 대표제품소개, 대표서비스소개
- 2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
- 3. 활용방법 : 각 샘플페이지에 대한 콘텐츠를 새로 구성하여 활용할 수 있다.

■ 고객지원

- 1. 세부메뉴 : 자료실, 묻고답하기
- 2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
- 3. 활용방법 : 각 샘플페이지에 기능을 추가개발후 구성하여 활용할 수 있다.

■ 알림마당

- 1. 세부메뉴 : 오늘의행사, 금주의행사, 공지사항, 사이트갤러리
- 2. 기능설명 : 공통컴포넌트 일정관리(부서일정)과 게시판 기능을 커스터마이징하여 사용한다.
- 3. 활용방법 : 관리자가 등록한 일정정보를 조회하거나 게시물을 조회할 수 있다.

■ 사이트관리

- 1. 세부메뉴 : 오늘의행사, 금주의행사, 공지사항, 사이트갤러리
- 2. 기능설명 : 공통컴포넌트 일정관리(부서일정)과 게시판 기능을 커스터마이징하여 사용한다.
- 3. 활용방법 : 관리자로 로그인 한 후 일정정보를 등하거나 게시물을 등록할 수 있다.(게시판 설정 가능)

내부업무 템플릿

■ 내부업무 템플릿관련 기본사항

- 1. 내부업무 템플릿은 업무시스템의 내부적인 관리기능(접속현황 모니터링, 접속제한, 사용자 및 권한 관리 등)을 위주로 기본 구성한 형태이다. 최초 관리자계정 설정은 [로그인계정 : admin , 로그인암호 : 1] 로 설정되어 있다.
- 2. 관리자 계정으로 업무용 사용자(예:user1 계정)을 추가등록하고 권한정보를 부여하는 방식으로 사용할 수 있고 기본 관리기능에 대한 초기 설정을 변경하여 적용할 수 있다.
- 3. 기본기능이나 예시메뉴를 실무적으로 추가 커스터마이징하여 활용할 수 있다.

■ 알림마당

- 1. 세부메뉴 : 공지사항, 업무게시판
- 2. 기능설명 : 공통컴포넌트 게시판 기능을 커스터마이징하여 사용한다.
- 3. 활용방법 : 관리자가 등록한 게시물을 조회할 수 있다.

■ 직급체계관리

- 1. 세부메뉴 : 입퇴사정보관리, 직급정보관리, 직위정보관리
- 2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
- 3. 활용방법 : 각 샘플페이지에 대한 콘텐츠를 새로 구성하여 활용할 수 있다.

■ 진급관리

1. 세부메뉴 : 업무평가접수 관리, 상벌정보 관리
2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
3. 활용방법 : 각 샘플페이지에 기능을 추가개발후 구성하여 활용할 수 있다.

■ 근태관리

1. 세부메뉴 : 출퇴근정보 관리, 휴무정보 관리 예시메뉴
2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
3. 활용방법 : 각 샘플페이지에 기능을 추가개발후 구성하여 활용할 수 있다.

■ 내부서비스관리

1. 세부메뉴 : 게시판관리(게시판템플릿관리,게시판생성관리,게시판사용여부관리,공지사항관리,업무게시판관리), 사용현황관리(접속로그관리,접속통계관리,로그인정책관리)
2. 기능설명 : 공통컴포넌트 게시판 기능, 접속로그, 접속통계, 로그인정책관리를 커스터마이징하여 사용한다.
3. 활용방법 : 관리자로 로그인 한 후 게시물을 등록할 수 있다.(게시판 설정 가능) 접속로그관련 정보를 모니터링 할 수 있다. 접속 계정들에 대한 개별 로그인정책을 조회할 수 있다.(접속제한)

■ 내부시스템관리

1. 세부메뉴 : 사용자관리(사용자등록관리,사용자부재관리), 사용자권한관리(권한관리,그룹관리,사용자별권한관리,롤관리), 메뉴관리(프로그램관리, 메뉴관리, 메뉴생성관리), 코드관리(분류코드관리,공통코드관리,상세코드관리)
2. 기능설명 : 공통컴포넌트 사용자관리, 권한관리(권한,그룹,사용자별권한,롤관리), 메뉴관리(프로그램,메뉴,메뉴생성), 코드관리(분류코드, 공통코드,상세코드) 기능을 커스터마이징하여 사용한다.
3. 활용방법 : 관리자로 로그인 한 후 게시물을 등록할 수 있다.(게시판 설정 가능) 사용자와 사용자권한정보를 관리할 수 있다. 프로그램과 메뉴정볼르 관리할 수 있다. 내부 코드정보를 관리할 수 있다.

포털사이트 템플릿

■ 포털사이트 템플릿관련 기본사항

1. 포털사이트 템플릿은 웹시스템의 커뮤니케이션 기능을 (게시판, FAQ, Q&A, 설문, 사용자 및 권한관리 등)을 위주로 기본 구성한 형태이다. 최초 관리자계정 설정은 [로그인계정 : admin , 로그인암호 : 1] 로 설정되어 있다.
2. 일반 유저(예:user1 계정)를 가입신청하고 관리자가 권한정보를 부여하는 방식으로 사용할 수 있고 FAQ, Q&A 등에 대한 추가 콘텐츠를 등록하여 사용 할 수 있다.
3. 프로그램, 메뉴에 대한 변경기능은 추가구성되어 있지 않으므로 필요시 DB를 통해 직접 변경한다.(내부업무 템플릿의 샘플 파일을 참조하여 재구성 후 업로드)
4. 기본기능이나 예시메뉴를 실무적으로 추가 커스터마이징하여 활용할 수 있다.

■ 사이트소개

1. 세부메뉴 : 사이트소개,연혁,조직소개,찾아오시는길
2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
3. 활용방법 : 각 샘플페이지에 대한 콘텐츠를 새로 구성하여 활용할 수 있다.

■ 민원마당

1. 세부메뉴 : 민원신청, 민원결과확인
2. 기능설명 : 예시메뉴에 해당하는 항목으로 샘플페이지 형태로 존재한다.
3. 활용방법 : 각 샘플페이지에 대한 콘텐츠를 새로 구성하여 활용할 수 있다.

■ 알림마당

1. 세부메뉴 : 공지사항, 자유게시판
2. 기능설명 : 공통컴포넌트 게시판 기능을 커스터마이징하여 사용한다.
3. 활용방법 : 관리자가 등록한 게시물을 조회할 수 있다.

■ 정보마당

1. 세부메뉴 : FAQ, Q&A, 설문조사
2. 기능설명 : 공통컴포넌트 FAQ, Q&A, 설문 기능을 커스터마이징하여 사용한다.
3. 활용방법 : 관리자가 등록한 게시형 콘텐츠정보를 조회할 수 있다.

■ 포털서비스관리

1. 세부메뉴 : 설문관리(설문지,설문템플릿,설문문항,설문항목,설문응답결과),서비스관리(FAQ관리,QA관리,게시판템플릿관리, 게시판생성관리, 게시판사용관리,공지사항관리,자유게시판관리,배너관리)
2. 기능설명 : 공통컴포넌트 설문 기능, FAQ, Q&A, 게시판, 배너 관리를 커스터마이징하여 사용한다.
3. 활용방법 : 관리자로 로그인 한 후 게시형 콘텐츠 정보를 등록할 수 있다.(게시판 설정 가능)

- 포털시스템관리

1. 세부메뉴 : 사용자관리(회원관리,이용약관관리,개인정보보호관리),사용자권한관리(권한관리,그룹관리,사용자별권한관리,롤관리)
2. 기능설명 : 공통컴포넌트 사용자관리, 권한관리(권한,그룹,사용자별권한,롤관리) 기능을 커스터마이징하여 사용한다.
3. 활용방법 : 사용자와 사용자권한정보를 관리할 수 있다.