**QMRA Application setup**

**Watershare tools**



Auteur: Marco Kortooms  
Versie: 0.5  
Documentdatum: 23 December 2013

network  
to final results

Inhoud

[1 Introduction 3](#_Toc375575965)

[2 Development environment 4](#_Toc375575966)

[3 Watershare Web Application 5](#_Toc375575967)

[3.1 Application layout 5](#_Toc375575968)

[Appendix A – Pom.xml 7](#_Toc375575969)

[Appendix B – spring-servlet.xml 9](#_Toc375575970)

[Appendix C – Web.xml 11](#_Toc375575971)

e for residual signing for cleaner water

# Introduction

This document contains an explanation of the application layout of the QMRA Watershare tool that is used as an example application for the Devlab project. The necessary components are is described in this document.

Watershare tool development environment consists of the next components:

* Development environment (Eclipse Kepler)
* Servlet/JSP server (Apache Tomcat) for dev testing
* Code repository (Git/GitHub)

The QMRA Watershare tool is a web application make use of the next components

* Components and Building using Maven and Git/Github
* Persistency/Database using Hibernate and HSQLDB
* Application framework using Spring
* Presentation Layout using JSP and JQuery
* Styling and Design using CSS

The layout of the application is described in section 3.

The QMRA tool is designated to run on an Apache Tomcat servlet/JSP server. The current version of QMRA has been tested on Apache Tomcat version 7.0.32.

# Development environment

To develop the QMRA Watershare applications Eclipse Kepler is used. This version of eclipse is available at the next location:

<http://www.eclipse.org/downloads/packages/eclipse-standard-431/keplersr1>

Standard GIT support is included in Eclipse Kepler. Additional plugins are not needed.

Maven integration and Spring support is available by installing the next plug-ins using the Eclipse Marketplace:

* JBoss Tools (Kepler)
* Spring source

It is recommended to import the code style and compiler preferences stored in the EclipsePreferences.epf file included into the ‘doc’-folder. This to make sure every contributor is using the same code style conventions in the applications.

# Watershare Web Application

As usual for Maven a list of dependencies is included in the pom.xml file. As groupId for every Watershare tool, KWRWatershare is selected, where the name of the tool is used as artifactId. Please make sure that the artifactId is lower case due to the fact that the artifactId is also used as application context.

An example of a pom.xml file is included in this document as appendix A.

## Application layout

The application code for QMRA Treatment Calculator is qmra.

The Watershare application layout is always:

|  |  |
| --- | --- |
| **File/Folder** | **Description** |
| src/main/java |  |
| nl.watershare.common | Classes for general us in all Watershare applications. |
| nl.watershare.common.security | Classes to check if the application is started using a valid portal session. |
| nl.watershare.dao | Package containing the Data Access Object classes for general use in all Watershare applications. |
| nl.watershare.entity | Package containing the Hibernate Data entity classes for general use in all Watershare applications. |
| nl.watershare.qmra.controller | Package containing the web controller classes for the Watershare application. |
| nl.watershare.qmra.doa | Package containing the Data Access Object classes for the Watershare application. These classes should implement the persistency to the database using Hibernate HQL code |
| nl.watershare.qmra.entity | Package containing the Hibernate Data entity classes for the Watershare application. In most cases this package will only contain a reference to the tables. |
| nl.watershare.qmra.form | Package containing the Spring form classes and Spring form validators for the Watershare application. |
| nl.watershare.qmra.form.types | Package containing miscellaneous Enumeration classes used to create form select boxes and used to store a default string value in de the DB tables. |
| nl.watershare.qmra.result | Package containing classes to create the result objects needed in the presentation layer |
| nl.watershare.qmra.result.criticalCondition | Package containing the classes to create the critical condition objects need in the presentation layer |
| nl.watershare.qmra.result.graph | Package containing the classes to create the graph objects needed in the presentation layer. |
| nl.watershare.qmra.service | Package containing the service classes as a link between the functionality (in forms) and the persistency for the Watershare application. |
| nl.watershare.qmra.utils | Package containing utility classes for the Watershare application. |
| nl.watershare.service | Package containing the service classes as a link between the functionality (in forms) and the persistency for general use in all Watershare applications. |
| src/main/resources |  |
| hibernate.cfg.xml | Hibernate configuration file to link entity classes to the database tables |
| qmra\_messages.properties | Translation file containing the Spring translations for the messages/labels for the Watershare application. |
| src/test/java | JUnit source code directory. Not yet used |
| src/test/resources | JUnit resource directory. Not yet used |
| src/main/webapp |  |
| css | CSS directory |
| fonts | Fonts directory |
| img | Images directory |
| js | Javascript directory. This directory contains JQuery API, Fancy Field API script files.  It also contains the general js-scripts for the application. These scripts are contained in the file ‘ functions.js’ |
| META-INF |  |
| WEB-INF |  |
| jsp | JSP directory containing unrendered JSP pages |
| Include |  |
| head.jsp | Header jsp file containing references to all the javascript/css files for the entire application. The head.jsp file will be loaded for every page. |
| title.jsp | Title jsp file containg the title image including the title of the application |
| error.jsp | Standard error file used by Spring to display errors. |
| spring-servlet.xml | Spring configuration file. See appendix B for an example |
| web.xml | Web configuration file. See appendix C for an example |
| pom.xml | Maven pom file |

# Appendix A – Pom.xml

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>KWRWatershare</groupId>

<packaging>war</packaging>

<version>1.0.1</version>

<description></description>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.6</source>

<target>1.6</target>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.jdom</groupId>

<artifactId>jdom</artifactId>

<version>2.0.2</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>4.3.0.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>3.6.9.Final</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.6.6</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.6.6</version>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>commons-pool</groupId>

<artifactId>commons-pool</artifactId>

<version>20030825.183949</version>

</dependency>

<!-- Gson: Java to Json conversion -->

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.2.1</version> <!-- makesure correct version here -->

</dependency>

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>org.hsqldb</groupId>

<artifactId>hsqldb</artifactId>

<version>2.2.8</version>

</dependency>

</dependencies>

<properties>

<org.springframework.version>3.1.2.RELEASE</org.springframework.version>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<artifactId>qmratreatmentcalculator</artifactId>

</project>

# Appendix B – spring-servlet.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:aop=*"http://www.springframework.org/schema/aop"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"* xmlns:jee=*"http://www.springframework.org/schema/jee"*

xmlns:lang=*"http://www.springframework.org/schema/lang"* xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:util=*"http://www.springframework.org/schema/util"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd*

*http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee.xsd*

*http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd"*>

<context:annotation-config />

<mvc:annotation-driven />

<context:component-scan base-package=*"nl.watershare.qmra, nl.watershare.entity, nl.watershare.service, nl.watershare.dao"* />

<bean id=*"propertyConfigurer"* class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"locations"*>

<list>

<value>classpath:jdbc.properties</value>

<!-- generic KWR watershare property file -->

<value>classpath:watershare.properties</value>

</list>

</property>

</bean>

<util:properties id=*"sessionValidation"*>

<prop key=*"externalSession.validation.url"*>${externalSession.validation.url}</prop>

<prop key=*"session.invalid.redirect.url"*>${session.invalid.redirect.url}</prop>

<prop key=*"externalSession.validation.turnedOn"*>${externalSession.validation.turnedOn}</prop>

</util:properties>

<!-- Configure the multipart resolver -->

<bean id=*"multipartResolver"* class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"* />

<!-- Set locale to default EN -->

<bean id=*"localeResolver"* class=*"org.springframework.web.servlet.i18n.CookieLocaleResolver"*>

<property name=*"defaultLocale"* value=*"en"* />

</bean>

<bean id=*"jspViewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"viewClass"*

value=*"org.springframework.web.servlet.view.JstlView"* />

<property name=*"prefix"* value=*"/WEB-INF/jsp/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<bean id=*"messageSource"* class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<property name=*"basename"* value=*"classpath:qmra\_messages"* />

<property name=*"defaultEncoding"* value=*"UTF-8"* />

</bean>

<bean id=*"dataSource"* class=*"org.apache.commons.dbcp.BasicDataSource"* destroy-method=*"close"*>

<property name=*"driverClassName"*>

<value>${jdbc.driverClassName}</value>

</property>

<property name=*"url"*>

<value>${jdbc.databaseurl}</value>

</property>

<property name=*"username"*>

<value>${jdbc.username}</value>

</property>

<property name=*"password"*>

<value>${jdbc.password}</value>

</property>

<property name=*"validationQuery"*>

<value>${jdbc.validationQuery}</value>

</property>

<property name=*"testOnBorrow"*>

<value>${jdbc.testOnBorrow}</value>

</property>

</bean>

<bean id=*"sessionFactory"*

class=*"org.springframework.orm.hibernate3.LocalSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"* />

<property name=*"configLocation"*>

<value>classpath:hibernate.cfg.xml</value>

</property>

<property name=*"configurationClass"*>

<value>org.hibernate.cfg.AnnotationConfiguration</value>

</property>

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>${jdbc.dialect}</prop>

<prop key=*"hibernate.show\_sql"*>false</prop>

</props>

</property>

</bean>

<tx:annotation-driven />

<bean id=*"transactionManager"* class=*"org.springframework.orm.hibernate3.HibernateTransactionManager"*>

<property name=*"sessionFactory"* ref=*"sessionFactory"* />

</bean>

</beans>

# Appendix C – Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* id=*"WebApp\_ID"* version=*"2.5"*>

<display-name>QMRA Treatment Calculator</display-name>

<welcome-file-list>

<welcome-file>start.do</welcome-file>

</welcome-file-list>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/spring-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>nl.watershare.utils.SetSystemPropertyListener</listener-class>

</listener>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<listener>

<listener-class>nl.watershare.utils.HSQLDBStartupListener</listener-class>

</listener>

<listener>

<listener-class>nl.watershare.common.security.SessionRegistry</listener-class>

</listener>

<filter>

<filter-name>SessionValidationFilter</filter-name>

<filter-class>nl.watershare.common.security.SessionValidationFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>SessionValidationFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<servlet>

<servlet-name>spring</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>\*.json</url-pattern>

</servlet-mapping>

<error-page>

<error-code>404</error-code>

<location>/WEB-INF/jsp/error.jsp</location>

</error-page>

<error-page>

<exception-type>java.lang.Exception</exception-type>

<location>/WEB-INF/jsp/error.jsp</location>

</error-page>

</web-app>