



iBit

Illes Balears
innovació
tecnològica

Registre Web

Documentació tècnica





Informació general del document.

Descripció.

Títol: Documentació tècnica Registre Web
Estat: Esborrany/Aprovat
Versió: 1.0.1
Autor/s: Toni Rogel
Creat: 23/11/09
Modificat 16/03/10
Fitxer: Documentacio Tecnica Regweb.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
Actualització detalls flags de configuració	Toni Rogel	15/07/10

Font documental.



Índice de contenido

1. Introducción	4
2. Instalación.....	4
2.1. Base de Datos.....	4
2.2. DataSource.....	4
2.3. Compilación.....	5
2.4. Autenticación en JBoss.....	5
3. Plugins.....	6
4. Estructura.....	6
4.1. ear	6
4.2. ejb.....	6
4.3. model.....	7
4.4. logic.....	7
4.5. webapp.....	7
5. Personalización.....	7
5.1. Plantillas.....	7
5.2. CSS.....	8
5.3. Imágenes.....	8
5.4. Textos Web.....	8
5.5. Opciones de configuración para compilación.....	8
5.6. Opciones de configuración en tiempo de ejecución.....	9
6. Modelo Entidad-Relación.....	10



1. Introducció

Se ha desarrollado este proyecto utilizando como base la máquina virtual de Java v.1.5.0-19 aunque se ha mantenido la compatibilidad con la versión 1.4.2-21. Como motor de base de datos se ha utilizado tanto DB2 7 como MySQL v5.0.95. Como servidor de aplicaciones se ha utilizado la versión de JBoss proporcionada por la DGTIC (3.2.8-caib7).

2. Instalación

2.1. Base de Datos

De cara a realizar la instalación, se proporciona un script SQL para crear la base de datos en MySQL.

Se incluye la posibilidad de generar el script SQL para la base de datos deseada. Para ello, hay que entrar en la carpeta *logic*, y modificar el fichero *config.properties* para indicar el dialecto de *Hibernate* deseado, tal y como se muestra a continuación:

Base de Datos	Dialecto Hibernate
MySQL	org.hibernate.dialect.MySQLDialect
Oracle	org.hibernate.dialect.Oracle9iDialect org.hibernate.dialect.Oracle10gDialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect
DB2	org.hibernate.dialect.DB2Dialect org.hibernate.dialect.DB2400Dialect

Una vez modificado este fichero, invocando el comando de Maven *mvn hibernate3:hbm2ddl* se generará el script SQL en la carpeta *logic/target/schema.ddl*.

2.2. DataSource

En el caso de JBOSS, para definir el DataSource hay que realizar los siguientes pasos:

1. Copiar los drivers de la base de datos en la carpeta *server/default/lib* de JBOSS.
2. Crear el fichero *regweb-ds.xml* o añadir la entrada del DataSource en el fichero *xxx-ds.xml* existente.



```
<?xml version="1.0" encoding="UTF-8"?>
<!--===== -->
<!-- -->
<!-- JBoss Server Configuration -->
<!-- -->
<!-- ===== -->
<datasources>
  <local-tx-datasource>
    <jndi-name>es.caib.regweb.db</jndi-name>
    <connection-url>jdbc:mysql://localhost/fipreg</connection-url>
    <driver-class>org.gjt.mm.mysql.Driver</driver-class>
    <user-name>root</user-name>
    <password></password>
  </local-tx-datasource>
</datasources>
```

2.3. Compilación

Para realizar la compilación de la aplicación se requiere la herramienta *Maven*, que se puede descargar desde la página <http://maven.apache.org>. Las últimas pruebas se han realizado con la versión 2.2.0.

Para compilar la aplicación se debe ejecutar la instrucción de Maven *mvn package* que descargará las dependencias, compilará y empaquetará la aplicación de cara a su despliegue.

Una vez compilada, se puede encontrar la aplicación empaquetada en la carpeta *ear/target*.

Así mismo, se pueden invocar las instrucciones *mvn idea:idea* o *mvn eclipse:eclipse*, que generarán respectivamente un proyecto para IntelliJ IDEA o Eclipse. En este caso, se recomienda ejecutar previamente la instrucción *mvn install* para garantizar un mejor funcionamiento de la generación de los proyectos.

2.4. Autenticación en JBoss

De cara a configurar la autenticación en JBoss, la aplicación utiliza el perfil *RWE_USUARI* para autenticar a los usuarios estándar, mientras que los administradores tienen el perfil *RWE_ADMIN*. Además, está el rol *RWE_LOPD* que tiene acceso a los logs generados para la LOPD. Independientemente del perfil que se tenga, una vez creadas las oficinas un administrador tiene que dar permisos a cada usuario para cada oficina con la que tenga que trabajar.



3. Plugins

Enmarcado en el ámbito de la tarea de desacoplamiento de las llamadas a funciones de COBOL, se ha desarrollado una interfaz de plug-in para permitir configurar otras funciones a realizar tras el alta, modificación o visado de un registro de entrada o salida, lo cual se ha aprovechado para implementar las llamadas existentes en la versión de la DGTIC.

Para ello se proporciona un proyecto independiente, llamado *regweb-module*, que ha de contener la implementación de una clase llamada `es.caib.regweb.module.PluginHook`, que se compone de dos funciones: *entrada* y *salida*. Actualmente los datos necesarios se pasan individualmente como parámetros. En la carpeta *regweb-module* se encuentra también un proyecto de maven que permite la compilación simplemente invocando el comando `mvn package`. Al igual que en el caso del proyecto principal, se encontrará la aplicación empaquetada en la carpeta *target*.

A su vez, si dicho proyecto requiere de un fichero de propiedades, debe estar en un paquete *.sar* llamado *regweb_module.sar*, que contenga estas propiedades. Se proporciona una implementación de ejemplo de este paquete.

4. Estructura

El código fuente está estructurado en los siguientes módulos:

4.1. ear

Contiene las configuraciones para realizar el empaquetado final de la aplicación con todas sus dependencias; el ejecutable compilado se encontrará en la carpeta *target*. Los ficheros *application.xml* y *jboss-app.xml* se generan automáticamente a partir de la configuración encontrada en el fichero *pom.xml*.

4.2. ejb

Contiene la antigua API de la aplicación, con *StatefulSessionBeans*, para mantener la compatibilidad con las aplicaciones existentes. El código fuente se encuentra en *src/main/java* mientras que los ficheros de configuración de los EJB *ejb-jar.xml* y *jboss.xml* se encuentran en la carpeta *src/main/resources*.



4.3. model

Contiene los modelos de las tablas usadas por la aplicación, utilizando anotaciones JPA. El código fuente se encuentra en la carpeta *src/main/java*.

4.4. logic

Contiene la nueva API de la aplicación, desarrollada utilizando Hibernate. Está estructurado de la siguiente forma:

- *config.properties*: configuración del dialecto de Hibernate deseado para la base de datos utilizada.
- *src/main/java*: código fuente del módulo. Utiliza XDoclet para la generación de las interfaces de los EJB y los ficheros XML de configuración a partir de un único fichero fuente.
- *src/main/resources/hibernate.cfg.xml*: fichero configuración de Hibernate donde se declaran las clases del módulo *model* que se mapean a la base de datos y el nombre del datasource utilizado.

4.5. webapp

Contiene la capa de presentación de la aplicación. Está estructurado de la siguiente forma:

- *src/main/java*: código fuente de los servlets.
- *src/main/webapp*: ficheros HTML y JSP de las vistas de la aplicación, además de la configuración de la capa web presente en la carpeta *WEB-INF*.
- *src/main/resources*: fichero *messages.properties* con los textos usados por la aplicación
- *src/personalizacion*: ficheros que controlan el aspecto de la aplicación web (plantillas *sitemesh*, imágenes y ficheros *css*).

Cada uno de estos módulos contiene un proyecto de maven con su fichero *pom.xml* en el que se puede personalizar el funcionamiento de cada uno de ellos.

5. Personalización

En la carpeta *regweb/webapp/src/personalizacion* hay 3 carpetas con los ficheros necesarios para personalizar la apariencia de la aplicación. Estas carpetas se incluyen automáticamente en el *war* en tiempo de compilación.

5.1. Plantillas

En la carpeta *decorators* hay 3 ficheros con las plantillas implementadas usando la librería *sitemesh*



utilizadas por la aplicación:

- *decorador.jsp*: utilizado por el módulo principal de usuario de la aplicación.
- *popup.jsp*: utilizado para las ventanas que aparecen para seleccionar un registro de una lista
- *admin.jsp*: utilizado por el módulo de administración

5.2. CSS

En la carpeta *css* se encuentra el fichero *estilos.css* utilizado por ambas plantillas para aplicar un mismo diseño.

5.3. Imágenes

En la carpeta *imagenes* se encuentran las imágenes usadas por la aplicación (logos, botones...) que se referencian desde el fichero *estilos.css* o alguno de los *decorators*.

5.4. Textos Web

En la carpeta *regweb/webapp/src/main/resources* se encuentra el fichero *messages.properties* con los textos utilizados en la aplicación.

5.5. Opciones de configuración para compilación

A continuación se detallan las opciones de configuración presentes en el archivo *regweb/config.properties* que se deben modificar antes de generar el ejecutable de la aplicación:

Opción	Valor por defecto	Descripción
admin.view.traspasos	false	Muestra enlace “traspasos” en menú administrador
registro.entrada.view.preregistre	true	Muestra enlace en menú principal para cargar desde pre-registro; la URL se configura según el punto 5.6
registro.entrada.view.disquete_correo	false	Muestra campos referentes a número de disquete y número de correo en formularios de registros de entrada y salida
registro.entrada.view.registre012	false	Muestra campos referentes a municipio del 012 en formularios de registros de entrada y salida



oficios.config.tipo_documento	OF	Código de tipo de documento que se asignará a un oficio de remisión generado por la aplicación
oficios.config.idioma	3	Código de idioma que se asignará a un oficio de remisión generado por la aplicación
oficios.config.entidad	OFICINA	Código de la entidad remitente que se asignará a un oficio de remisión generado por la aplicación
webapp.welcomeFile	lanza.jsp	Permite seleccionar la página de inicio de la aplicación, si se desea elegir una distinta tras personalizar el aspecto
hibernate.dialect		Dialecto de Hibernate seleccionado para la base de datos; descrito en el punto 2.1

5.6. Opciones de configuración en tiempo de ejecución

Las siguientes opciones se deben configurar en un fichero *.sar* para permitir su configuración sin necesidad de recompilar la aplicación:

Opción	Descripción
es.caib.regweb.preregistre	URL del servidor donde se encuentra instalado SISTRA para el enlace a “Cargar desde pre-registro”

