**EEUtil-vis-docexp**

**Manual de Instalación**

*Documento de Integración*

**Sistemas Desarrollo**

*Versión* del documento

**0001**

*Fecha de revisión*

**31/10/2018**

*Realizado por*

**Sistemas Desarrollo**

**ÍNDICE**

[1 Control de modificaciones 3](#_Toc528762191)

[2 Objetivo de este documento 4](#_Toc528762192)

[3 Requisitos previos 5](#_Toc528762193)

[4 Instalación 6](#_Toc528762194)

[4.1 Base de datos 6](#_Toc528762195)

[Instalación de EEUtils (MySQL) 6](#_Toc528762196)

[4.2 Ficheros de configuración 7](#_Toc528762197)

[log4j.properties 7](#_Toc528762198)

[database.properties (MySQL) 7](#_Toc528762199)

[afirma.properties 7](#_Toc528762200)

[visualización.textos.properties 8](#_Toc528762201)

[4.3 Instalación de EEUtil-vis-docexp a partir del código fuente 9](#_Toc528762202)

[Instalación de librerías externas: 9](#_Toc528762203)

[Descarga de dependencias: 10](#_Toc528762204)

[Especificar versión del conector de base de datos MySQL: 10](#_Toc528762205)

[Ejecutar EEUtil-vis-docexp en entorno local 10](#_Toc528762206)

[Construcción del WAR a partir del código fuente 10](#_Toc528762207)

[4.4 Instalación de EEUtil-vis-docexp a partir del war. 11](#_Toc528762208)

[5 Alta de nuevas aplicaciones 12](#_Toc528762209)

[Tabla inside\_aplicaciones 12](#_Toc528762210)

[Tabla inside\_aplicaciones\_propiedad 12](#_Toc528762211)

[6 Web Services externos a EEUtil-vis-docexp 14](#_Toc528762212)

[7 Anexo Tomcat 14](#_Toc528762213)

[Instalación en Tomcat 14](#_Toc528762214)

# Control de modificaciones

Revisión Actual: 001

Fecha: 31/10/2018

Autor: DSIC - Sistemas Desarrollo

Descripción:

*-* Primera versión del manual

# Objetivo de este documento

Este documento pretende ser una guía de instalación del sistema EEUtil-vis-docexp.

Para conseguir el WAR (fichero de despliegue de aplicaciones Web) existen dos alternativas:

* + Descargarlo directamente.
  + Realizar la compilación del código fuente y posterior empaquetamiento.

Si se ha descargado el WAR directamente puede ignorarse el punto 4.3

# Requisitos previos

Para poder realizar una instalación correcta y completa de EEUtil-vis-docexp, es necesario que el sistema cumpla los siguientes requisitos:

* + Base de datos: MySQL. Versión 5.5.9. **La aplicación no es compatible con la versión 5.6 de MySQL.**
  + **JDK 1.7** o superior.
  + Servidor de aplicaciones Apache Tomcat (versión 6 o superior), o similar.
  + Apache Maven 2 (si se desea trabajar con el código fuente).

# Instalación

En este apartado se detallan los pasos para realizar la instalación y configuración correcta de EEUtil-vis-docexp, tanto a través del war como del código fuente.

## Base de datos

En este apartado se describen los pasos para realizar la instalación de EEUtil-vis-docexp.

### Instalación de EEUtils (MySQL)

En primer lugar (tanto para el war como para el código fuente) es necesario crear las tablas y objetos necesarios para el funcionamiento de EEUtils. Los scripts se encuentran tanto en la carpeta de código fuente como en la de los binarios. Podemos encontrarlos en la ruta resources/scripts\_bbdd.

Existen varios scripts que incluye creación de tablas y los datos mínimos necesarios que estas deben llevar para poder arrancar la aplicación:

* + 01-create-tables.sql
  + 02-inserts.sql
  + 03-create\_procedures.sql
  + 04-insert-prueba-test-user.sql
  + 05-insert-admin-eeutil-misc.sql

Estos scripts son comunes a todos los proyectos de EEUtils, por lo que deben ejecutarse una única vez.

Se incluye el usuario “prueba” con la contraseña “test” para realizar pruebas.

Para crear nuevos usuarios, modifique el script 04-insert-prueba-test-user.sql con los datos del nuevo usuario.

El script 05-insert-admin-eeutil-misc.sql debe ser modificado para incluir los identificadores de los certificados con el que se vayan a logar en caso de utilizar eeutil-misc.

## Ficheros de configuración

Los ficheros de configuración se proporcionan en la ruta resources/config y pueden colocarse en la ruta que se desee. Además, este directorio debe indicarse en el arranque de la aplicación, asignando la ruta a la variable de entorno eeutil-vis-docexp.config.path:

java ….. -Deeutil-vis-docexp.config.path=<ruta\_ficheros\_configuracion>

También debe especificarse esta misma ruta para la variable local\_home\_app:

java ….. -Dlocal\_home\_app=<ruta\_ficheros\_configuracion>

### log4j.properties

En este fichero se configurarán las propiedades de log de la aplicación. El que se proporciona en la distribución contiene necesita ser configurado en la siguiente propiedad, para establecer la ruta en la que dejar el fichero de log:

log4j.appender.GENERAL.File=/XXX/eeutil-vis-docexp.log

### database.properties (MySQL)

En este fichero se configuran las propiedades de la base de datos.

database.driverClassName=com.mysql.jdbc.Driver

database.url=jdbc:mysql://host:port/database

database.username=XXX

database.password=XXX

database.showSql=0

database.hibernateDialect=org.hibernate.dialect.MySQLDialect

No se proporciona ningún driver por lo que deberán incluir el driver especificado según la versión de su MySQL en las librerías del servidor de aplicaciones.

### afirma.properties

En este fichero se configuran las propiedades para consumir los servicios web de @firma:

# PARÁMETROS PARA LA CONEXIÓN CON @FIRMA

afirma.validarfirma.url=http://afirma.redsara.es/afirmaws/services/ValidarFirma

afirma.validarcertificado.url=http://afirma.redsara.es/afirmaws/services/ValidarCertificado

afirma.verifydss.url=http://afirma.redsara.es/afirmaws/services/DSSAfirmaVerify

afirma.idaplicacion=XXX

afirma.user=XXX

afirma.password=XXX

afirma.truststore=/XXX/truststore.jks

afirma.truststore.password=XXX

afirma.idaplicacionServidorDedicado=XXX

afirma.connectionTimeOut=120000

afirma.receiveTimeOut=120000

Se deben especificar las direcciones donde se encuentran los web services de @firma a utilizar en las variables afirma.validarfirma.url, afirma.validarcertificado.url y afirma.verifydss.url.

Se deben especificar los datos de autenticación de @firma en las variables afirma.idaplicacion, afirma.user y afirma.password.

La ruta del truststore se especifica en afirma.truststore y la contraseña del mismo en afirma.truststore.password.

### visualización.textos.properties

En este fichero se configuran los textos que se mostrarán en la visualización.

visualizacion.textos.documento.pagina.texto=Página

visualizacion.textos.documento.cabecera=Visualización de Documento Electrónico

visualizacion.textos.documento.titulo=Documento Electrónico

visualizacion.textos.documento.tipo.firma=Tipo Firma

visualizacion.textos.documento.fecha.firma=Fecha de Firma

visualizacion.textos.documento.firmante=Firmante

visualizacion.textos.documento.valor.csv=Valor CSV

visualizacion.textos.documento.csv=CSV

visualizacion.textos.documento.regulacion.csv=Regulación CSV

visualizacion.textos.documento.indice=Documento Índice

visualizacion.textos.expediente.indice=ÍNDICE DE EXPEDIENTE ELECTRÓNICO

visualizacion.textos.expediente.cabecera=Expediente Electrónico

visualizacion.textos.comunes.elemento=ELEMENTO

visualizacion.textos.comunes.Hash=HASH

## Instalación de EEUtil-vis-docexp a partir del código fuente

El código fuente de EEUtil-vis-docexp se distribuye en distintos módulos:

* + eeutil-mvn-base: Proyecto “padre”, no contiene código fuente, tan solo las propiedades y dependencias comunes.
  + eeutil-afirmaws-stub: Módulo que consume los servicios de @firma.
  + eeutil-comunes: Módulo que da soporte a otros módulos y que contiene interceptores SOAP, capa de seguridad, definición de excepciones de servicios web...
  + eeutil-inside-stub: Módulo que consume los servicios web de inside.
  + eeutil-model: Módulo que da soporte a otros módulos y que contiene clases POJO comunes.
  + eeutil-pdf-conversion: Módulo que da soporte sobre la conversión de documentos a formato pdf utilizando la librería de iText.
  + eeutil-pdf-conversion-igae: Módulo que da soporte sobre la conversión de documentos a formato pdf compatible con las necesidades de IGAE.
  + eeutil-services: Módulo que da soporte sobre clases DAO, servicios de aplicaciones y usuarios, rutas de configuración...
  + eeutil-util: Módulo que da soporte sobre certificados, archivos, mimetypes, xml...
  + load-tables:Módulo que da soporte a las transacciones con la base de datos.
  + eeutil-vis-docexp: Proyecto principal

### Instalación de librerías externas:

Para el desarrollo de EEUtil-vis-docexp a partir del código fuente es necesaria la inclusión de librerías externas que no se encuentran publicadas en los repositorios públicos de Maven. Estas librerías se encuentran en la carpeta resources/external-libs:

* + afirma
  + igae
  + jodconverter

Dentro de cada una de estas carpetas se incluyen las librerías necesarias y un documento install\_jars.txt con los comandos de Maven necesarios para la instalación de cada una de ellas en nuestro repositorio local.

### Descarga de dependencias:

La instalación de dependencias se puede realizar instalando los proyectos en el siguiente orden:

* + eeutil-afirmaws-stub
  + eeutil-comunes
  + eeutil-inside-stub
  + eeutil-model
  + eeutil-pdf-conversion
  + eeutil-pdf-conversion-igae
  + eeutil-services
  + eeutil-util
  + load-tables
  + eeutil-mvn-base
  + eeutil-vis-docexp

### Especificar versión del conector de base de datos MySQL:

En la raíz del proyecto eeutil-vis-docexp se encuentra el fichero pom.xml. Se debe buscar mysql-connector-java en el fichero y, si es necesario, cambiar la versión del conector de base de datos por el que corresponda a la versión de MySQL a la que se va a atacar.

### Ejecutar EEUtil-vis-docexp en entorno local

Una vez se han rellenado correctamente los ficheros de configuración, descargado las dependencias y compilado el código, la aplicación puede ser ejecutada en un entorno de servidor local, ejecutando la siguiente orden en el directorio eeutil-vis-docexp del proyecto:

mvn -Deeutil-vis-docexp.config.path=<ruta\_ficheros\_configuracion> -Dlocal\_home\_app=<ruta\_ficheros\_configuracion> clean jetty:run

Se puede deducir que la aplicación ha arrancado correctamente si en la consola no aparecen errores y si se puede acceder desde un navegador a la dirección:

Muestra la lista con los servicios web disponibles:

[http://](http://localhost:8090/inside)localhost:8096/eeutil-vis-docexp/ws

### Construcción del WAR a partir del código fuente

Para crear el WAR a partir del código fuente se ejecuta la siguiente orden en cada uno de los proyectos y en el orden indicado:

mvn clean package install

* + eeutil-afirmaws-stub
  + eeutil-comunes
  + eeutil-inside-stub
  + eeutil-model
  + eeutil-pdf-conversion
  + eeutil-pdf-conversion-igae
  + eeutil-services
  + eeutil-util
  + load-tables
  + eeutil-mvn-base
  + eeutil-vis-docexp

Finalmente, para el proyecto eeutil-vis-docexp se ejecutará:

mvn -P homovivus package war:war

En el directorio target del proyecto eeutil-vis-docexp se habrá creado el fichero de despliegue eeutil-vis-docexp.war, que contendrá dentro todas las librerías necesarias para la ejecución de la aplicación.

## Instalación de EEUtil-vis-docexp a partir del war.

El war sumistrado por el distribuible no contiene la librería de conexión de la base de datos, por tanto, se debe incluir en las librerías del servidor de aplicaciones el conector adecuado según la versión de la Base de Datos de MySQL a la que se va a atacar.

# Alta de nuevas aplicaciones

Las aplicaciones consumidoras de los Web Services de EEUtil-vis-docexp deberán estar dadas de alta en las tablas inside\_aplicaciones e inside\_aplicaciones\_propiedad.

La gestión de estas tablas debe realizarse directamente en la base de datos o bien desde la aplicación eeutil-misc.

En los scripts de generación de la base de datos se incluye en el documento 04-insert-prueba-test-user.sql un ejemplo con las sentencias SQL INSERT a ejecutar para crear un usuario. A continuación se describen los campos de las tablas.

### Tabla inside\_aplicaciones

* + **idaplicacion**: identificador de la aplicación.
  + **password**: hash del password de la aplicación, calculado con el algoritmo md5.
  + **activo**: Indica si la aplicación está activa, si su valor es false no le dejará realizar ninguna operación
  + **descripcion**: nombre o descripción de la aplicación.
  + **activa**: Indica si la aplicación puede está activa.
  + **tramitar**: Indica si la aplicación puede realizar tareas de trámites.
  + **sello**: Indica si la aplicación puede realizar tareas de sello electrónico.
  + **firma**: Indica si la aplicación puede realizar tareas de firma.
  + **email**: Email del responsable de la aplicación.
  + **telefono**: Teléfono de contacto del responsable de la aplicación.
  + **responsable**: Nombre del responsable de la aplicación.
  + **unidad**: DIR3 de la unidad orgánica a la que pertenece la aplicación.

### Tabla inside\_aplicaciones\_propiedad

* + **idaplicacion:** identificador de la aplicación.
  + **propiedad:** nombre de la propiedad.
  + **valor:** valor de la propiedad.

El valor que puede tomar la columna **propiedad** puede ser los siguientes:

* + **algoritmoFirmaDefecto:** algoritmo de firma que se utilizará por defecto
  + **formatoFirmaDefecto:** formato de la firma que se utilizará por defecto
  + **modoFirmaDefecto:** modo de la firma que se utilizará por defecto
  + **rutaKS:** ruta donde se encuentra el keystore a utilizar. Se recomienda hacer uso de la variable ${local\_home\_app} para que los keystores se encuentren en la misma ruta que los ficheros de configuración.
  + **tipoKS:** tipo del keystore especificado en la propiedad rutaKS.
  + **passwordKS:** password para poder acceder al keystore especificado en rutaKS.
  + **aliasCertificado:** alias del certificado a utilizar que se encuentra dentro del keystore especificado en rutaKS.
  + **passwordCertificado:** password del certificado especificado en aliasCertificado.
  + **ip.openoffice:** dirección IP donde se encuentra el servicio de OpenOffice.
  + **port.openoffice:** puerto donde se encuentra el servicio de OpenOffice.
  + **rutaLogo:** ruta donde se encuentra el logotipo a utilizar en la estampación de los documentos. Se recomienda hacer uso de la variable ${local\_home\_app} para que el logotipo se encuentre en la misma ruta que los ficheros de configuración.

# Web Services externos a EEUtil-vis-docexp

Algunas funcionalidades de EEUtil-vis-docexp dependen de la implementación de Web Services externos. Estos Web Services tienen que cumplir una especificación determinada.

La especificación de estos servicios web se encuentra en la carpeta /Web-Services-Externos.

- **@firma**: Especificación de varios WS de validación para verificación y obtención de información de firmas y certificados electrónicos.

Se implementa un Cliente Web Service por cada una de las especificaciones establecidas por los ficheros WSDL que se encuentran en la ruta /Web-Services-Externos /afirma.

# Anexo Tomcat

Los ficheros de configuración se proporcionan en la ruta resources/config y pueden ubicarse en la ruta que se desee**, teniendo en cuenta se accederá a ellos mediante la variable ${local\_home\_app}**.

### Instalación en Tomcat

Esta es la instalación más sencilla de la aplicación. Es necesario un contenedor de servlets tipo Tomcat. Se recomienda en su versión 6 o mayor. Para llevar a cabo la instalación del war descargue la última versión del área de descargas.

Para el caso de Tomcat, el war puede ser instalado a través del manager de despliegue o simplemente ubicarlo en la carpeta de aplicaciones webapp. Busque la forma si no se trata de Tomcat.

En la carpeta /lib debe ubicar la librería con el conector de base de datos de la versión adecuada a la versión de MySQL que ataque en caso de utilizar directamente el war proporcionado por el distribuible.

Por último, queda indicar las variables de configuración de EEUtil-vis-docexp. Recomendamos que para dichos parámetros se utilice la variable de entorno JAVA\_OPTS, de forma que esté disponible al ejecutar Tomcat:

|  |
| --- |
| JAVA\_OPTS="$JAVA\_OPTS -Dlocal\_home\_app=<ruta\_ficheros\_configuracion> -Deeutil-vis-docexp.config.path<ruta\_ficheros\_configuracion>" |