**CSVStorage – Manual de Instalación**

Almacenamiento de CSVs

*Documento de Integración*

**Sistemas Desarrollo**

*Versión* del documento

**001**

*Fecha de revisión*

**23/10/2018**

*Realizado por*

**Sistemas Desarrollo**

**ÍNDICE**

1 Control de modificaciones 3

2 Objetivo de este documento 4

3 Requisitos previos 5

4 Instalación 6

4.1 Base de datos 6

Instalación de CSVStorage (Oracle) 6

4.2 Ficheros de configuración 6

log4j.properties 6

datasource.properties (Oracle) 7

eeutil.properties 7

csvstorage.properties 8

dir3.properties 8

ginside.properties 8

truststore.jks 9

ws-security.properties 9

Directorio schemas 9

4.3 Instalación de CSVStorage a partir del código fuente 10

Descarga de dependencias: 10

Ejecutar CSVStorage en entorno local 10

Construcción del WAR a partir del código fuente 10

5 Web Services externos a CSVStorage 12

6 Anexo Tomcat 12

Instalación en Tomcat 12

# Control de modificaciones

# Objetivo de este documento

Este documento pretende ser una guía de instalación del sistema CSVStorage.

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 CSVStorage, es necesario que el sistema cumpla los siguientes requisitos:

* + Base de datos: Oragle. Versión 10G.
  + **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 CSVStorage, tanto a través del war como del código fuente.

## Base de datos

En este apartado se describen los pasos para realizar la instalación de CSVStorage.

### Instalación de CSVStorage (Oracle)

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

Existe un único script que incluye creación de tablas y los datos mínimos necesarios que estas deben llevar para poder arrancar la aplicación:

* + BDcsvstorage.sql

Incluye el usuario admin con password admin dado de alta como usuario administrador.

La password codificado hash SHA-256

## Ficheros de configuración

Los ficheros de configuración se proporcionan en la ruta resources/config deben guardarse en el mismo directorio. Además, este directorio debe indicarse en el arranque de la aplicación, asignando la ruta a la variable de entorno config.path:

java ….. -Dconfig.path=<ruta\_ficheros\_configuracion>

### log4j.properties

En este fichero se configuraran las propiedades de log de la aplicación. El que se proporciona en la distribución contiene tres *appender:* consola, fichero y correo. No obstante podrá configurarse de otro modo si así se desea. Para una instalación inicial de pruebas se recomienda eliminar los *appenders* file y mail y dejar únicamente stdout, por lo que el fichero quedaría de la siguiente forma:

# Root logger option

log4j.rootLogger= INFO, stdout

log4j.logger.es.gob.aapp= INFO, stdout

log4j.additivity.es.gob.aapp=false

# Direct log messages to stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern= ${LOG\_PATTERN}

### datasource.properties (Oracle)

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

database.driver= oracle.jdbc.driver.OracleDriver

database.url=jdbc:oracle:thin://host\:port/database

database.username=XXX

database.password=XXX

jpa.generate-ddl = false

jpa.hibernate.ddl-auto = none

jpa.show-sql = false

jpa.database-platform = org.hibernate.dialect.OracleDialect

jpa.database=ORACLE

El driver proporcionado se ha probado para la versión 10G de Oracle. Si se desea instalar CSVStorage sobre una versión de Oracle no compatible con este driver deberá incorporarse el driver adecuado a las dependencias del proyecto.

Los campos a rellenar son los siguientes:

* + **database.driver**: Driver de conexión. Si se modifica se deberá aportar el jar correspondiente a las dependencias del proyecto.
  + **database.url**: URL de conexión a la BBDD.
  + **database.username**: Usuario de conexión a la BBDD.
  + **database.password**: Password de conexión a la BBDD.

### eeutil.properties

Para que CSVStorage pueda validar los ENIs que recibe, se deberá implementar un Web Service que cumpla una especificación determinada. Ver punto 0

En el fichero eeutil.properties se configuran las propiedades de acceso a dicho Web Service:

eeutil.misc.ws.ruta =http://host:port/eeutil-misc/ws/EeUtilValidacionENIServiceMtom?wsdl

eeutil.misc.ws.idaplicacion =xxx

eeutil.misc.ws.password=xxx

Los campos a rellenar son:

* + **eeutil.misc.ws.ruta:** URL del Web Service.
  + **eeutil.misc.ws.idaplicacion:** Identificador de la aplicación CSVStorage en el servicio.
  + **eeutil.misc.ws.password:** Password de la aplicación CSVStorage en el servicio.

### csvstorage.properties

En el siguiente fichero se configura el almacenamiento de los documentos que recibe CSVStorage:

sistemaficheros.ruta = ./docs

url.consultar.sistema.referencia = http://HOST:PORT/csvstorage/services/CSVQueryDocumentWSService

Los campos a rellenar son:

* + **sistemaficheros.ruta**: url donde almacenar los documentos.
  + **url.consultar.sistema.referencia**: URL del Web Service que la posibilidad a otras aplicaciones de consultar documentos de las que no son propietarias.

### dir3.properties

Para la carga de datos de maestros en CSVStorage (unidades orgánicas procedentes de DIR3), es necesario rellenar el fichero de dir3.properties.

Ejemplo de propiedades:

dir3.ws.SD01UNDescargaUnidades.url.wsdl= http://pru-dir3ws.redsara.es/directorio/services/SD01UN\_DescargaUnidades?wsdl

dir3.ws.usuario =XXX

dir3.ws.password =XXX

dir3.ws.ruta.descarga.zip=./build/tmp/dir3\_tmp/

dir3.ws.ruta.descarga.unzip=./build/tmp/dir3\_tmp/unzipped/

dir3.ws.nombre.xml.unidades= Unidades.xml

En el fichero se especifican las siguientes propiedades:

* + **dir3.ws.SD01UNDescargaUnidades.url.wsdl:** url para obtener los datos básicos de unidades orgánicas procedentes del DIR3.
  + **dir3.ws.usuario:** usuario para acceder al servicio web del Dir3 para unidades orgánicas.
  + **dir3.ws.password:** password para acceder al servicio web del Dir3 para unidades orgánicas.
  + **dir3.ws.ruta.descarga.zip:** url donde se descargará el contenido en formato zip
  + **dir3.ws.ruta.descarga.unzip:** url donde se descomprime el zip descargado anteriormente.
  + **dir3.ws.nombre.xml.unidades:** nombre final del fichero que contiene los Dir3 descargados.

### ginside.properties

Para generar documentos ENIs, se realizará por medio de los servicios proporcionados por SIA, para ello es necesario rellenar el fichero ginside.properties.

Ejemplo del fichero:

ginsidemtom.ws.usertoken.url.wsdl=http://HOST:PORT/inside/ws/GInsideUserTokenMtomService?wsdl

ginside.ws.usertoken.url.wsdl= http://HOST:PORT/inside/ws/GInsideUserTokenService?wsdl

ginsidemtom.ws.idaplicacion =XXX

ginsidemtom.ws.password =XXX

ginside.ws.idaplicacion=XXX

ginside.ws.password=XXX

En el fichero se especifican las siguientes propiedades:

* + **ginsidemtom.ws.usertoken.url.wsdl:** url del servicio web para generar ENI.
  + **ginside.ws.usertoken.url.wsdl:** url del servicio web para generar ENI.
  + **ginsidemtom.ws.idaplicacion:** usuario para acceder al servicio web del mtom.
  + **ginsidemtom.ws.password:** password para acceder al servicio web del mtom.
  + **ginside.ws.idaplicacion:** usuario para acceder al servicio web del usertoken.
  + **ginside.ws.password:** password para acceder al servicio web del usertoken.

### truststore.jks

Almacén de claves facilitado vacío cuya password para abrirlo es “password”.

Deberá incluir el certificado proporcionado por cl@ve, que incluye una clave privada y una pública para la generación del token de petición de acceso al servicio.

### ws-security.properties

Configuración de seguridad para realizar peticiones firmadas. Se usa en conjunción con este fichero y los almacenes de claves trustStore.jks

Ejemplo del fichero:

schema-validation-enabled=false

ws-security.enable.nonce.cache= false

ws-security.enable.timestamp.cache=false

#Propiedaes de configuración del WSS4J

org.apache.wss4j.crypto.provider= org.apache.wss4j.common.crypto.Merlin

org.apache.wss4j.crypto.merlin.truststore.file= ./src/config/local/truststore.jks

org.apache.wss4j.crypto.merlin.truststore.password=xxx

org.apache.wss4j.crypto.merlin.truststore.type= JKS

### Directorio schemas

Bajo este directorio deben almacenarse todos los esquemas XSD del documento para la correcta validación de éstos.

* + ContenidoDocumentoEni.xsd
  + CSVQueryDocument.xsd
  + CSVQueryDocumentCredential.xsd
  + CSVQueryDocumentWSS.xsd
  + DocumentoEni.xsd
  + firmasEni.xsd
  + MetadatosDocumentoEni.xsd
  + ReferenciaDocumento.xsd
  + xmldsig-core-schema.xsd

## Instalación de CSVStorage a partir del código fuente

El código fuente de CSVStorage se distribuye en un solo proyecto:

* + Scvstorage.

### Descarga de dependencias:

La descarga de las librerías e instalación de dependencias se debe realizar instalando el proyecto:

* + csvstorage

Nota: Para que compile el proyecto, así como para que su ejecución se realice de forma correcta es necesario que la JRE tenga instalada la librería que se encuentra en el directorio external-libs. Esta librería se proporciona en la carpeta resources/external-libs y es:

* + bigDataTransfer-1.0.jar

### Ejecutar CSVStorage 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 clase Application.java

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:

Entrar en la aplicación web:

<http://localhost:8080/>csvstorage

Para mostrar una lista con los servicios disponibles:

<http://localhost:8080/csvstorage>/services

### Construcción del WAR a partir del código fuente

Para crear el WAR a partir del código fuente se ejecuta la siguiente orden en el proyecto:

gradlew clean war

* + csvstorage

En el directorio libs del proyecto csvstorage se habrá creado el fichero de despliegue csvstorage.war, que contendrá dentro todas las librerías necesarias para la ejecución de la aplicación (exceptuando las librerías que se tienen que colocar en el directorio endorsed de la JRE).

# Web Services externos a CSVStorage

Algunas funcionalidades de CSVStorage dependen de la implementación de Web Services externos. Estos Web Services tienen que cumplir una especificación determinada.

- **firma-remota**: Especificación de WS de firma remota.

Para que los documentos y el índice de los expedientes puedan ser firmados en servidor (sellados) se ha de implementar un Web Service que cumpla con el WSDL que se encuentra en la ruta Web Services externos/firma-remota. También se proporciona en este directorio el esquema XSD con la definición de los tipos de entrada y de salida.

- **eeutil**: Especificación de WS de validación de documentos xml.

- **inside**: Especificación de WS para generar documentos ENIs.

- **dir3**: Especificación de WS de carga de tablas. Se encarga de rellenar la tabla de unidades orgánicas.

# Anexo Tomcat

Los ficheros de configuración se proporcionan en la ruta resources/config **deben ubicarse todos en el mismo directorio**. Será importante el lugar en el que se ubique, ya que, dicha ruta tendrá que utilizarse para establecer el valor del parámetro de configuración obligatorio de CSVStorage “**config.path”**.

### 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.

Por último queda indicar las variables de configuración de CSVStorage. 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 -Dconfig.path=<<ruta carpeta aplicacion>>/conf" |