



Plataforma @firma

Cliente Custodia - Manual de integración

Versión: v01r05

Fecha: 26/01/2016

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito de la Junta de Andalucía.

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
---	--	---

HOJA DE CONTROL

Título	Cliente Custodia - Manual de integración		
Entregable	Documentación técnica Plataforma @firma		
Nombre del Fichero	20160126-Manual Integrador cliente custodia_v01r05		
Autor	DGPD		
Versión/Edición	v01r05	Fecha Versión	26/01/2016
Aprobado por		Fecha Aprobación	DD/MM/AAAA
		Nº Total Páginas	13

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Área	Fecha del Cambio
v01r00	Creación del documento	UTE	UTE	12/04/2013
v01r01	Corrección de errores de formato	UTE	UTE	18/04/2013
v01r02	Eliminado método storeDocument	UTE	UTE	15/07/2013
v01r03	Se indica la URL del repositorio de artefactos Maven de CHAP en el apartado 3.3	UTE	UTE	12/05/2014
v01r04	Configuración avanzada	UTE	UTE	21/07/2014
v01r04	(afirma-custody-client-1.0.3) Se permite el uso del almacén de confianza por defecto de la JRE (apartado 3.4)	UTE	UTE	26/01/2016

	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
---	--	---

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Área	Nº Copias
Manuel Perera Domínguez	Jefe de Servicio	CHAP	1
José Ignacio Cortés Santos	Director de Proyecto	CHAP	1
Antonio Heredia Rizo	Consultor	UTE	1

ÍNDICE

1 Introducción	5
2 Servicios nativos de Custodia de @firma.....	6
2.1 Catálogo de servicios.....	6
3 Integración con afirma-custody-client.....	7
3.1 Dependencias software.....	7
3.2 Dependencias de servicios.....	7
3.3 Establecer dependencias con Maven.....	8
3.4 Fichero de propiedades.....	9
3.5 Interfaz del componente afirma-custody-client.....	10
3.6 Ejemplos de integración.....	11
4 Configuración avanzada.....	13

	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
---	---	--

1 Introducción

El componente afirma-custody-client tiene como objetivo fundamental facilitar a los integradores la interacción con los servicios básicos del módulo de Custodia disponibles en la plataforma @firma. Se trata de un componente software que implementa la lógica necesaria relacionada con la mensajería y las comunicaciones (SOAP), ofreciendo a los integradores una interfaz sencilla.

afirma-custody-client está desarrollado en java y está construido con Maven siguiendo las recomendaciones de MADEJA.

Internamente, afirma-custody-client se divide en dos partes funcionales:

- Por un lado, realiza la composición de la petición y el procesamiento de la respuesta XML implementando la lógica de mensajería.
- Por otro lado, el cliente incluye la implementación del envío y recepción del mensaje SOAP que contiene la petición y respuesta respectivamente, por lo que este proceso es transparente para el integrador. El cliente hace uso de librerías de código abierto como Axis y Web Service Security para realizar el intercambio de información con el servicio web de @firma.

Este manual describe con detalle la interfaz proporcionada por el componente afirma-custody-client, cómo se configura y cómo se integra con las aplicaciones.

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
---	---	--

2 Servicios nativos de Custodia de @firma

2.1 Catálogo de servicios

En este apartado se detallan todos los Servicios Web publicados por la plataforma @Firma para el módulo de Custodia:

- **DeleteDocumentContent:** Servicio que permite la eliminación de un documento, almacenado en el módulo de Custodia por @Firma, a partir de su identificador de documento.
- **GetBlock:** Servicio que permite la obtención de un bloque de firmas, almacenado en el módulo de Custodia por @Firma.
- **GetDocId:** Servicio que permite la obtención del identificador de un documento custodiado en @Firma, a partir del identificador de transacción de la firma del documento.
- **GetDocumentContent:** Servicio que permite la obtención del contenido de un documento almacenado en el módulo de Custodia, a partir del identificador de transacción de la firma del documento.
- **GetDocumentContentByDocId:** Servicio que permite la obtención del contenido de un documento almacenado en el módulo de Custodia, a partir de su identificador de documento.
- **GetESignature:** Servicio que permite la obtención de una firma custodiada en @Firma, a partir de su identificador de transacción.
- **GetTransactionsByApplId:** Servicio que permite la obtención de la lista completa de transacciones de firma registradas por una aplicación determinada.
- **GetTransactionsByDates:** Servicio que permite la obtención de una lista de transacciones de firma, registradas por una aplicación determinada, en un rango de fechas concreto.
- **GetTransactionsByRefId:** Servicio que permite la obtención de una lista de transacciones de firma a partir de una referencia externa.
- **SetExternalReference:** Servicio que permite actualizar la referencia externa de una transacción de firma.
- **StoreDocument:** Servicio que permite el almacenamiento de un documento en el módulo de Custodia de @Firma.

De todos los servicios descritos, sólo los subrayados están implementados por el componente afirma-custody-client.

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
--	--	---

3 Integración con afirma-custody-client

3.1 Dependencias software

El componente afirma-custody-client está implementado en java y compilado con JDK 5. Son necesarias una serie de dependencias (ficheros jar) que deben estar en el CLASSPATH de la aplicación que hace uso del componente. La librería que implementa el cliente se denomina *afirma-custody-client-1x.x.jar* y las dependencias mencionadas son las siguientes:

- commons-logging:commons-logging:jar:1.1
- log4j:log4j:jar:1.2.12
- logkit:logkit:jar:1.0.1
- avalon-framework:avalon-framework:jar:4.1.3
- javax.servlet:servlet-api:jar:2.3
- commons-lang:commons-lang:jar:2.1
- org.apache.ws.commons.util:ws-commons-util:jar:1.0.2
- xml-apis:xml-apis:jar:1.0.b2
- commons-codec:commons-codec:jar:1.6
- xerces:xercesImpl:jar:2.4.0
- axis:axis:jar:1.4
- org.apache.axis:axis-jaxrpc:jar:1.4
- org.apache.axis:axis-saaj:jar:1.4
- axis:axis-wsdl4j:jar:1.5.1
- commons-discovery:commons-discovery:jar:0.2
- org.apache.ws.security:wss4j:jar:1.6.7
- org.apache.santuario:xmlsec:jar:1.5.2
- org.opensaml:opensaml:jar:2.5.1-1
- org.opensaml:openws:jar:1.4.2-1
- org.opensaml:xmltooling:jar:1.3.2-1
- org.slf4j:slf4j-api:jar:1.6.1
- joda-time:joda-time:jar:1.6.2

3.2 Dependencias de servicios

Para el correcto funcionamiento de afirma-custody-client es necesario que los servicios web de @firma 5 sean visibles desde el servidor donde se pretende ejecutar el componente (host y puerto).

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
--	--	---

3.3 Establecer dependencias con Maven

El componente `afirma-custody-client` puede ser incluido en aplicaciones desarrolladas en maven añadiendo la siguiente dependencia al `pom.xml` de la aplicación:

```
<dependency>
  <groupId>es.juntadeandalucia.afirma</groupId>
  <artifactId>afirma-custody-client</artifactId>
  <version>1.x.x</version>
</dependency>
```

Este componente, y todos los desarrollados por la Consejería de Hacienda y Administración Pública pueden encontrarse en el Repositorio Maven del Servicio de Coordinación de Administración Electrónica (<https://ws024.juntadeandalucia.es/ae/adminelec/repositoriomaven>).

Para que Maven obtenga el artefacto que contiene el componente y resuelva automáticamente sus dependencias, es necesario incluir el siguiente fragmento XML en el fichero `pom.xml` correspondiente:

```
<repositories>
  <repository>
    <id>SCAERepo</id>

    <url>https://ws024.juntadeandalucia.es/maven/repository/internal</url>
  </repository>
</repositories>
```

NOTA: El Repositorio Maven del Servicio de Coordinación de Administración Electrónica de la Consejería de Hacienda y Administración Pública está accesible a través del protocolo HTTPS utilizando el certificado **cn=*.juntadeandalucia.es, ou=FNMT Clase 2 CA, o=FNMT, c=ES** de la Junta de Andalucía. Para que Maven confíe en el dominio **ws024.juntadeandalucia.es** a través de HTTPS y logre descargar el componente y sus dependencias, es necesario que incluya el certificado indicado en el almacén de certificados por defecto (`cacerts`) de la Máquina Virtual Java sobre la que se ejecute el script **mvn**.

3.4 Fichero de propiedades

Para configurar correctamente el componente afirma-custody-client, es necesario establecer algunos parámetros de configuración en un fichero de propiedades denominado **afirma.properties**. Dicho fichero de propiedades debe incluirse en el CLASSPATH de la aplicación que utilice el componente.

```
#####
# Configuración del componente afirma-custody-client
#####

#####
# Credenciales de la aplicación (obligatorio)
#
# Si se establecen las propiedades de autenticación mediante BinarySecurityToken
# (certificado) este tipo de autenticación prevalece respecto a la autenticación
# mediante UsernameToken (usuario / contraseña).
#####
afirma.idapp = STERIA_TEST
#####

#####
# BinarySecurityToken
#####

# afirma.authorization.ks.path = chap.pl2

# afirma.authorization.ks.type = PKCS12
# afirma.authorization.ks.password = chap
# afirma.authorization.ks.cert.alias = chap
#####

#####
# UsernameToken
#####
afirma.user = user
afirma.password = pass
#####

#####
# Datos del servidor (obligatorio)
#
# El componente siempre realizará las peticiones al siguiente endpoint:
# https://<afirma_host>/afirmaws/services/<nombre_del_servicio>
#####
afirma.host = ws028.juntadeandalucia.es
#####

#####
# Almacén de certificados de confianza (opcional)
#
# El componente presupone que el almacén de certificados de confianza está
# incluido en el CLASSPATH de la aplicación, por lo que en 'afirma.truststore'
# debe indicarse el nombre del fichero JKS (o bien, la rula relativa al mismo
# dentro del CLASSPATH). Si se establece el valor "default" se hace uso del
# almacén de confianza de la JRE que ejecuta la aplicación (cacerts)
```

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
--	--	---

```
#
# Por ejemplo, una aplicación web (empaquetada en un WAR) cuyo almacén está
# situado en el directorio WEB-INF/classes/almacenconfianza.jks, deberá
# establecer la propiedad afirma.truststore de la siguiente manera:
#
# afirma.truststore = almacenconfianza.jks
#
# También se admite referenciar al almacén de confianza mediante su ruta absoluta.
#
#####
afirma.truststore = trust.jks
afirma.truststorePassword = storepass
#####
```

3.5 Interfaz del componente afirma-custody-client

En este apartado se describe el interfaz del componente afirma-custody-client. Para más información acerca de las clases de respuesta puede consultar el Javadoc del componente.

NOTA: El ámbito de visibilidad de los métodos de obtención de información del módulo de Custodia está restringido al ID de aplicación configurado en la propiedad **afirma.idapp**, de manera que las aplicaciones sólo podrán obtener sus documentos y firmas propios.

3.5.1 getDocumentContent

Servicio para recuperar un documento almacenado en el módulo de Custodia de @Firma.

- Parámetros de entrada:
 - transactionId: Identificador de la transacción de la firma del documento.
- Respuesta:
 - Objeto de tipo GetDocumentContentResponse con los datos de la respuesta.
- Cabecera del método:


```
public GetDocumentContentResponse getDocumentContent( String transactionId )
throws CustodyException;
```

3.5.2 getDocumentContentByDocId

Servicio para recuperar un documento almacenado en el módulo de Custodia de @Firma.

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
--	--	---

- Parámetros de entrada:
 - docId: Identificador del documento custodiado.
- Respuesta:
 - Objeto de tipo `GetDocumentContentResponse` con los datos de la respuesta.
- Cabecera del método:


```
public GetDocumentContentResponse getDocumentContentByDocId( String docId )
    throws CustodyException;
```

3.5.3 getESignature

Servicio para recuperar una firma almacenada en el módulo de Custodia de @Firma.

- Parámetros de entrada:
 - transactionId: Identificador de transacción de una firma custodiada en @Firma.
- Respuesta:
 - Objeto de tipo `GesESignatureResponse` con los datos de la respuesta.
- Cabecera del método:


```
public GesESignatureResponse getESignature( String transactionId ) throws
    CustodyException;
```

3.6 Ejemplos de integración

A continuación se muestran ejemplos de integración de cada uno de los servicios implementados por el componente.

3.6.1 Recuperación de un documento (getDocumentContent)

```
// Creo una instancia del componente
CustodyClient custodyClient = CustodyClientBuilder.getCustodyClient();

// Invoco el servicio GetDocumentContent
GetDocumentContentResponse response = custodyClient.getDocumentContent( "1365674096609848" );
```

Como es de esperar, el identificador “1365674096609848” debe ser un identificador de transacción válido y existente en la plataforma.

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
--	---	--

3.6.2 Recuperación de un documento (getDocumentContentByDocId)

```
// Creo una instancia del componente
CustodyClient custodyClient = CustodyClientBuilder.getCustodyClient();

// Invoco el servicio GetDocumentContent
GetDocumentContentResponse response = custodyClient.getDocumentContentByDocId( "227902" );
```

Como es de esperar, el identificador “227902” debe ser un identificador de documento válido y existente en la plataforma.

3.6.3 Recuperación de una firma (getESignature)

```
// Creo una instancia del componente
CustodyClient custodyClient = CustodyClientBuilder.getCustodyClient();

// Invoco el servicio GetESignature
GetESignatureResponse response = custodyClient.getESignature( "1365674096609848" );
```

Como es de esperar, el identificador “1365674096609848” debe ser un identificador de transacción válido y existente en la plataforma.

 JUNTA DE ANDALUCÍA	Consejería de Hacienda y Administración Pública Dirección General de Política Digital	Plataforma @firma Cliente Custodia - Manual de integración
--	--	---

4 Configuración avanzada

Con el objetivo de flexibilizar la configuración del componente `afirma-custody-client`, éste permite ser configurado opcionalmente mediante un objeto de la clase **`java.util.Properties`**. Dicho objeto debe contener las propiedades de forma análoga a como se definen en el fichero `afirma.properties` (3.4 Fichero de propiedades). A modo de ejemplo, se indica a continuación como debe invocarse el componente con esta configuración:

```
/*
 * Definición del fichero de propiedades
 */
Properties afirmaProperties = new Properties();
afirmaProperties.put( "afirma.idapp", "IDAPP" );
afirmaProperties.put( "afirma.user", "user" );
afirmaProperties.put( "afirma.password", "pass" );
afirmaProperties.put( "afirma.host", "ws028.juntadeandalucia.es" );
afirmaProperties.put( "afirma.truststore", "trustStore.jks" );
afirmaProperties.put( "afirma.truststorePassword", "passTrustStore" );
// Pueden definirse el resto de propiedades incluidas en el fichero afirma.properties

// Invocación avanzada del componente
CustodyClient custodyClient = new CustodyClientImpl( new CustodyConfiguration().configure( afirmaProperties ) );

// Llamada a la operación
```