

Manual Programador API Document Custody

Guia per Programadors



G VICEPRESIDÈNCIA
O I CONSELLERIA
I INNOVACIÓ,
B RECERCA I TURISME
/ FUNDACIÓ BIT

Informació general del document.

Descripció.

Títol:	Com Usar Api Document Custody
Estat:	Esborrany/Aprovat
Versió:	1.0
Autor/s:	Antoni Nadal Bennasar
Creat:	30/05/2017
Modificat	06/06/2017
Fitxer:	ManualProgramadorApiDocumentCustody.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
Primera versió	A. Nadal	30/05/2017

Font documental.

Index de Contingut

1.-Introducció.....	4
2.-Com usar aquesta API.....	4
2.1.-Repositoris.....	4
2.2.-Dependències.....	5
2.2.1.-Dependència en el mòdul on es faci ús.....	5
2.2.2.-Dependència en el mòdul “ear”.....	5
2.3.-Propietats de selecció del Plugin.....	6
2.4.-Propietats associades al Plugin.....	6
2.5.-Codi per a fer la Custòdia.....	7
2.5.1.-Inicialitzar DocumentCustody.....	7
2.5.2.-Reserva CustodyID.....	7
2.5.3.-Crear PDF, estampar i signar.....	8
2.5.4.-Guardar Signatura en Custòdia.....	9
3.-Mètodes i Beans de l'API.....	10
4.-Plugins de Document Custody.....	10
4.1.-FileSystem.....	10
4.2.-Arxiu Digital CAIB.....	11
4.3.-Alfresco.....	11
4.4.-Alfresco APB.....	11

IMPORTANT: Accedir a

<https://github.com/GovernIB/pluginsib/tree/pluginsib-1.0/plugins-documentcustody/doc> per comprovar si hi ha una nova versió d'aquest document.

1.- Introducció

El present document intenta explicar als programadors com usar l'API de Document Custody en els programes a desenvolupar per la CAIB o Altres Entitats.

Segueix la filosofia de definir una API i diverses implementacions que anomenarem Plugins que implementen l'API. D'aquesta forma podem modificar i/o actualitzar les implementacions sense ni haver de modificar o recompilar l'aplicació. Per més detalls d'aquesta filosofia d'API-Plugins podeu consultar els documents d'aquesta carpeta: <https://github.com/GovernIB/pluginsib/tree/pluginsib-1.0/doc>.

API de Document Custody serveix per emmagatzemar documents electrònics. Un document electrònic està format per un fitxer, una firma i un conjunt de metadades. L'API opcionalment permet metadades i annexes.

És important comentar que aquesta API no s'ha d'usar com a sistema general per guardar tot tipus de documents.

2.- Com usar aquesta API

A continuació es descriuen les passes per incloure

2.1.- Repositoris

S'ha d'afegir al pom pare el següent repositori, per poder descarregar les APIs i els Plugins.

```
<repositories>

  <repository>
    <id>maven-repos</id>
    <name>GitHub GovernIB Maven Repository</name>
    <url>http://GovernIB.github.io/maven/maven/</url>
  </repository>

</repositories>
```

2.2.- Dependències

2.2.1.- Dependència en el mòdul on es faci ús

En el mòdul on s'hagi de de usar la custòdia, s'ha d'incloure la següent dependency:

```
<dependency>
  <groupId>org.fundaciobit.plugins</groupId>
  <artifactId>plugin-documentcustody-api</artifactId>
  <version>[[VERSIO]]</version>
  <scope>provided</scope>
</dependency>
```

La versió actual és la 3.0.0, però s'ha de consultar a l'entitat per la que es desenvolupi quina versió és la que s'ha d'usar.

2.2.2.- Dependència en el mòdul “ear”

Inclou la següent dependència dins el mòdul de l'ear

```
<dependency>
  <groupId>org.fundaciobit.plugins</groupId>
  <artifactId>plugin-documentcustody-api</artifactId>
  <version>[[VERSIO]]</version>
</dependency>
```

Per a desenvolupar s'utilitzarà el Plugin de FileSystem:

```
<dependency>
  <groupId>org.fundaciobit.plugins</groupId>
  <artifactId>plugin-documentcustody-filesystem</artifactId>
  <version>[[VERSIO_FS]]</version>
</dependency>
```

Nota: La versió VERSIO_FS actualment és la 3.0.0 però s'ha de consultar quina versió usar.

Cada Entitat, en producció i preproducció usarà el Plugin definit en les seves normatives de programació. Per exemple, en aplicacions dirigides a la CAIB s'usarà el Plugin de Document Custody d'Arxiu Digital CAIB:

```
<dependency>
  <groupId>org.fundaciobit.plugins</groupId>
  <artifactId>plugin-documentcustody-arxiudigitalcaib</artifactId>
  <version>[[VERSIO_AD]]</version>
</dependency>
```

Nota: La versió VERSIO_AD actualment és la 3.0.2 però s'ha de consultar quina versió usar.

Al final d'aquest document s'expliquen la resta de Plugins i les seves dependències (Veure punt 4.-Plugins de Document Custody)

2.3.- Propietats de selecció del Plugin

L'API en si mateixa no fa res, només és un conjunt de mètodes buits, beans i constants. Per això per fer feina "hem d'associar" una implementació a l'API. La forma més senzilla de fer això és definir una propietat que ens indicui la classe del Plugin que volem emprar. Aquesta propietat pot estar en bbdd, en un fitxer de configuració, en el <<app>>-properties-services.xml de l'aplicació, ... però en s'ha de poder canviar en qualsevol moment sense haver de recompilar l'aplicació.

A partir d'aquí emprem el fitxer <<app>>-properties-services.xml per definir aquestes i altres propietats necessàries per configurar el sistema de Document Custody. També emprem un package d'exemple: es.caib.exemple que en les vostres aplicacions heu de canviar pel vostre package. La propietat que en definirà el plugin a empra és la següent:

El valor d'aquesta classe per cada Plugin la podeu trobar al punt 4.-Plugins de Document Custody). En l'exemple següent usam el Plugin de FileSystem per guardar els document electrònics:

```
es.caib.exemple.plugindecustodia=  
org.fundaciobit.plugins.documentcustody.filesystem.FileSystemDocumentCustodyPlugin
```

En desenvolupament és sol utilitzar elPlugin de FileSystem i ja en Pre i Pro ja es canviaria pel Plugin que utilitza l'entitat (Per exemple en el cas de CAIB l'Arxiu Digital CAIB. Veure punt i 4.2.-Arxiu Digital CAIB))

2.4.- Propietats associades al Plugin

Cada implementació de l'API (Plugin) necessita d'un conjunt de propietats per configurar el seu funcionament intern i també per oferir flexibilitat a les diferents necessitats de cada aplicació.

Aquestes propietats estan formades per tres parts:

- **Fixa**: és le package de l'aplicació. Seguint l'exemple seria es.caib.exemple
- **Comú del Plugin**: és la part comú que defineix el propi plugin. Per exemple pel plugin de File System tendriem: plugins.documentcustody.filesystem
- **Propietat**: És el nom de la propietat. Per exemple "basedir" defineix el directori on guardar el fitxers.

Les propietats és munten d'aquesta forma:

[FIXA][ESPECIFICA][PROPIETAT_KEY]=[VALUE]

Per exemple per definir el directori on guarda el Plugin de FileSystem els fitxers és muntaria

d'aquesta forma:

```
es.caib.exemple.plugins.documentcustody.filesystem.basedir=C:/directoriDeCustodia
```

Podeu veure les propietats completes del Plugin de FileSystem al punt 4.1.-FileSystem.

Com que no i pot haver col·lisió entre propietats a causa la part “Comú del Plugin”, podem tenir varies configuracions dins del fitxer <<app>>--properties-service.xml sense cap problemes. En les aplicacions de la CAIB ja que s'han inclòs els plugin de File System i Arxiu Digital CAIB, també s'haurien d'afegir dins d'aquest fitxer les propietats associades a aquests dos plugins (veure punt 4.1.-FileSystem i 4.2.-Arxiu Digital CAIB)

Important: S'han de revisar totes les propietats de cada plugin i ajustar-les a les nostres necessitats. Per exemple la propietat basedir del plugin de FileSystem, s'han de fer que apunti a un directori correcte, si feim ús d'aquest plugin.

2.5.- Codi per a fer la Custòdia

Nota: recordau a canviar el package “es.caib.exemple” per el de la vostra aplicació.

2.5.1.- Inicialitzar DocumentCustody

```
String packageBase = "es.caib.exemple.";
String classname = System.getProperty(packageBase + "plugindecustodia");
IDocumentCustodyPlugin documentCustodyPlugin;
documentCustodyPlugin = (IDocumentCustodyPlugin)
PluginsManager.instancePluginByClass(classname, packageBase, System.getProperties());
```

La variable classname recollirà la propietat definida en el punt “2.3.-Propietats de selecció del Plugin” i les propietats del plugin seran recollides del System.getProperties() amb base fixa “packageBase” (veure cridada instancePluginByClass()).

2.5.2.- Reserva CustodyID

Abans de començar a guardar document, hem de fer el que es diu una reserva, que significa que el plugin ens retornarà un identificador amb el qual podem realitzar futures operacions sobre aquest Document Electrònic.

Certs plugins requereixen informació addicional que només es troba dins de les estructures de dades de cada aplicació, d'aquí que s'hagi de cercar algun mecanisme per passar aquesta informació al Plugin. La solució és el que anomenam “custodyParameters” que simplement és un map de claus String i valors Object.

Els valors que s'haurien de passar a través del `custodyParameters` serien els següents:

- Informació del funcionari/empleat públic que realitza l'acció.
- Informació del ciutadà pel qual es fa aquests guardat
- Informació del document: `validexDocumento`, tipus documental, perfil firma, estat elaboració,

En resum s'han de passar al Custody parameters les dades que poden variar entre diferents Document Electrònics.

Aquest pas d'informació es pot fer en un sol objecte o en varis i tant el nom com el bean passat pot tenir qualsevol estructura. Nota: En els exemple de l'Arxiu Digital i File System podem observar propietats “_EL” que signifiquen qu'epoden ser constants o expressions que és calculen a partir dels `custodyParameters` (veure punt 4.1.-FileSystem i 4.2.-Arxiu Digital CAIB i consultar web http://freemarker.org/docs/dgui_template_exp.html)

```
Map<String, Object> custodyParameters= new HashMapMap<String, Object>();
custodyParameters.put("infouser", <<BEAN INFO FUNCIONARIO: username i DNI >>);
custodyParameters.put("infociutada", <<BEAN INFO CIUTADA: nom i DNI >>); // Opcional
custodyParameters.put("infodoc", <<BEAN INFO DEL DOCUMENT >>); (*) (**)
String custodyID = documentCustodyPlugin.reserveCustodyID(custodyParameters);
```

Un cop obtingut el `custodyID`, l'hem de guardar normalment en BBDD per d'aquesta forma poder operar i consultar amb el Document Electrònic associat.

(*) Els camps d'aquest objecte han de definir:

- "estat_elaboracio" (EE01,EE02, EE03, EE04, EE99)
- "tipus Documental" (TD01, TD02, TD03, ...),
- origen_document (Administració 1 , Ciutadà 0)

(**) Si aquest programa gestiona diferent "procediments administratius" també ha d'incloure el codi del procediment administratiu.

<< FALTA : EXEMPLE DE CUSTODY PARAMETERS i DE PROPIETATS >>

2.5.3.- Crear PDF, estampar i signar

Com a exemple suposam que el fitxer a guardar és un PDF firmat al qual li hem d'estampar el CSV si en te. Podria ser perfectament una firma CADES o XADES i en aquest cas, podríem ometre aquest pas.


```
Metadata mcsv =
documentCustodyPlugin.getOnlyOneMetadata(custodyID,MetadataConstants.ENI_CSV);
String csv;
if(mcsv == null) {
    csv = "NO TE CSV";
} else {
    csv = mcsv.getValue();
}
String url = documentCustodyPlugin.getValidationUrl(custodyID, custodyParameters);
String specialValue =
documentCustodyPlugin.getSpecialValue(custodyID,custodyParameters);

byte[] pdf = crearPDF(dades, csv, url, specialValue);

byte[] pdf_firmat = firmarPDF(pdf);
```

2.5.4.- Guardar Signatura en Custòdia

```
SignatureCustody signatureCustody = new SignatureCustody();
signatureCustody.setName("prova.pdf");
signatureCustody.setData(pdf_firmat);
signatureCustody.setLenght(pdf_firmat.length);
signatureCustody.setMime("application/pdf");
signatureCustody.setAttachedDocument(true);
signatureCustody.setSignatureType(SignatureCustody.PADES_SIGNATURE);
// És un PDF i no té Arxiu Associat. El PDF ja conté l'arxiu i la firma
DocumentCustody documentCustody = null;
// No definim metadades addicionals
Metadata[] metadades = null;

documentCustodyPlugin.saveAll(custodyID, custodyParameters, documentCustody,
signatureCustody, metadades);
```

Nota: En el cas de XAdES i CAdES detached haurien de definir el DocumentCustody com segueix:

```
DocumentCustody documentCustody = new DocumentCustody();
documentCustody.setName("originaldoc.txt");
documentCustody.setData(data);
documentCustody.setMime("text/plain");
documentCustody.setLenght(data.length);
```

<< FALTA : EXEMPLE DE COM RECUPERAR UN DOCUMENT o SIGNATURA >>

3.- Mètodes i Beans de l'API

<< FALTA METODES I BEANS>>

4.- Plugins de Document Custody

A continuació es descriuen els detalls dels plugins disponibles.

4.1.- FileSystem

<< FALTA DEPENDENCIES I PROPIETATS>>

Nom	Descripció

Propietats:

```
<!-- ===== PLUGIN DE CUSTODIA - FILESYSTEM ===== -->

es.caib.exemple.documentcustodyplugin=org.fundaciobit.plugins.documentcustody.filesystem.FileSystemDocumentCustodyPlugin

es.caib.exemple.plugins.documentcustody.filesystem.basedir=D:/dades/dades/Proyectos/REGW
EB/Archivos/

# ===== Generiques AbstractDocumentCustodyPlugin (veure projecte /plugin-
documentcustody-api per a altres propietats)

# Informació de ExpressionLanguage: http://freemarker.org/docs/ref_directives.html

# Opcional. Servei per generar la URL de Custodia. Si no es defineix retornarà null.
# {0} => custodyID // {1} => URLEncode(custodyID) // {2} => Hash(custodyID)
es.caib.exemple.plugins.documentcustody.filesystem.baseurl=http://localhost:8080/annexos/index
.jsp?custodyID={1}
```

```
# Opcional. Configuració per generar Hash de la baseurl quan {2}  
es.caib.exemple.plugins.documentcustody.filesystem.hash.password=mypass  
# MD2, MD5, SHA, SHA-256, SHA-384, SHA-512  
es.caib.exemple.plugins.documentcustody.filesystem.hash.algorithm=MD5
```

4.2.- Arxiu Digital CAIB

Tots els detalls d'aquest plugin estan documentats dins dels document que es troben en aquesta adreça: <https://github.com/GovernIB/pluginsib/tree/pluginsib-1.0/plugins-documentcustody/arxiudigitalcaib/doc>.

4.3.- Alfresco

<< FALTA DEPENDENCIES I PROPIETATS>>

Nom	Descripció

4.4.- Alfresco APB

Nom	Descripció

<< FALTA DEPENDENCIES I PROPIETATS>>