Plugins de RegWeb3

Manual d'Integració Plugins de RegWeb3

GOIB VICEPRESIDÈNCIA I CONSELLERIA INNOVACIÓ, RECERCA I TURISME

GOIB VICEPRESIDÈNCIA I CONSELLERIA INNOVACIÓ, RECERCA I TURISME FUNDACIÓ BIT

Informació general del document

Descripció.

Títol:	Manual de Plugins Caib de RegWeb3
Estat:	Esborrany/Aprovat
Versió:	1.0
Autor/s:	Marilén González del Amo / Joan Pernía Bestard
Creat:	18/04/18
Modificat	05/12/19
Fitxer:	Manual_Integracio_Plugins_Regweb3.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
Modificat 2.2.- Plugin de Custòdia de Justificant	J.Pernía	23/04/18
Revisades urls dels plugins a Github	J.Pernía	20/05/19
Actualitzats url de documentació dels plugins externs de regweb3	M.González	05/12/19
Actualitzada explicació integració Regweb3-Plugins externs	M.González	18/02/20
Actualizades versions dels plugins externs que empra regweb3	M. González	25/05/21

Font documental.

Índex de Contingut

1.- Introducció

1.1.- Introducció General

RegWeb3 disposa d'una sèrie de plugins que permeten a cada entitat que implementi la funcionalitat pròpia. Aquest document explica quina és la funcionalitat que es pot implementar de cada un i com es configuren dins l'aplicació.

2.- Plugins propis de Regweb3

2.1.- Plugin de Distribució

El plugin de distribució ens permet distribuir els distints registres d'entrada cap a un conjunt de destinataris. El que se vulgui distribuir dependrà de la implementació que es faci de la interfície del plugin.

L'API del plugin està formada per les següents classes i interfície.

2.1.1.- Interfície IDistribucionPlugin

```
public interface IDistribucionPlugin extends IPlugin {

	public static final String DISTRIBUCION_BASE_PROPERTY = IPLUGIN_BASE_PROPERTIES +
"distribucion.";


    /**
     * Metodo que obtiene los destinatarios a los que se debe distribuir el registro de entrada.
     * @param registro registro de entrada que se distribuye
     * @return
     * @throws Exception
     */
    Destinatarios distribuir(RegistroEntrada registro) throws Exception;


    /**
     * Método que envia/distribuye el registro de entrada a la lista de destinatarios  indicada
     * @param registro registro de entrada que se distribuye
     * @param destinatariosDefinitivos destinatarios a los que enviar el registro de entrada
     * @param observaciones observaciones al envio
     * @return
     * @throws Exception
     */
    Boolean enviarDestinatarios(RegistroEntrada registro,
        List<Destinatario> destinatariosDefinitivos,
        String observaciones, Locale locale) throws Exception;


    /*
      Método que devuelve la configuración de la distribución.
     */
    ConfiguracionDistribucion configurarDistribucion() throws Exception;
}
```

El mètode distribuir ha de retornar el llistat de destinataris a on se vol enviar el registre.

El mètode enviarDestinatarios ha de fer tota la feina per enviar l'assentament a la llista de Destinataris que se li ha indicat i es pot afegir unes observacions i l'idioma.

2.1.2.- Classe Destinatario

```
public class Destinatario {

    private String id;

    private String name;

    public Destinatario() {
    }

    /**
     * @param id
     * @param name
     */
    public Destinatario(String id, String name) {
        this.id = id;
        this.name = name;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Destinatario [name=" + name + ", id=" + id + "]";
    }
}
```

Aquesta classe representa un destinatari a on s'enviaran les anotacions. Aquest destinatari està definit per un identificador(id) i un nom(name).

2.1.3.- Classe Destinatarios

```
public class Destinatarios {

    @XmlAttribute
    private List<Destinatario> propuestos;

    @XmlAttribute
    private List<Destinatario> posibles;

    public Destinatarios() {
    }

    /**
     * @param propuestos
     * @param posibles
     */
    public Destinatarios(List<Destinatario> propuestos, List<Destinatario> posibles) {
        this.propuestos = propuestos;
        this.posibles = posibles;
    }
```

```
    public List<Destinatario> getPropuestos() {
        return propuestos;
    }

    public void setPropuestos(List<Destinatario> propuestos) {
        this.propuestos = propuestos;
    }

    public List<Destinatario> getPosibles() {
        return posibles;
    }

    public void setPosibles(List<Destinatario> posibles) {
        this.posibles = posibles;
    }
}
```

Com s'ha explicat a l'apartat anterior un Destinatari ve representat per un id i un nom, però el plugin dona la possibilitat d'enviar un assentament a una llista de Destinataris. Per això la classe «Destinatarios» defineix dues llistes de Destinataris els possibles i els proposats pel sistema extern, de tal manera que l'usuari mitjançant una interfície podrà triar a quins destinataris finals enviar l'assentament.

2.1.4.- Classe ConfiguracionDistribucion

```
public class ConfiguracionDistribucion {
    /**
     * Configura si el usuario de registro  puede escoger o modificar el listado de destinatarios a
quien enviar el registro
     */
    public boolean listadoDestinatariosModificable;
    //especifica que información se enviará en el segmento de anexo del registro de entrada.
    /* 1 = custodiaId + metadades + fitxer + firma. És a dir a dins el segment annexes de
l'assentament s'enviaria tot el contingut de l'annexe.
    *  2 =  custodiaId. A dins el segment annexes de l'assentament només s'enviaria l'Id del sistema
que custodia l'arxiu.
    *  3 = custodiaId + metadades. A dins el segment annexes de l'assentament s'enviaria l'Id del
sistema que custodia l'arxiu i les metadades del document.
    * */
    public int configuracionAnexos;
    public int maxReintentos;
    public boolean envioCola;
    public ConfiguracionDistribucion(boolean listadoDestinatariosModificable, int
configuracionAnexos) {
        this.listadoDestinatariosModificable = listadoDestinatariosModificable;
        this.configuracionAnexos = configuracionAnexos;
    }
    public ConfiguracionDistribucion(boolean listadoDestinatariosModificable, int
configuracionAnexos,int maxReintentos, boolean envioCola) {
        this.listadoDestinatariosModificable = listadoDestinatariosModificable;
        this.configuracionAnexos = configuracionAnexos;
        this.maxReintentos = maxReintentos;
        this.envioCola = envioCola;
    }
    public boolean isListadoDestinatariosModificable() {
        return listadoDestinatariosModificable;
    }
    public void setListadoDestinatariosModificable(boolean listadoDestinatariosModificable) {
        this.listadoDestinatariosModificable = listadoDestinatariosModificable;
    }
    public int getConfiguracionAnexos() {
        return configuracionAnexos;
    }
    public void setConfiguracionAnexos(int configuracionAnexos) {
        this.configuracionAnexos = configuracionAnexos;
    }
    public int getMaxReintentos() {
        return maxReintentos;
    }
```

```
    public void setMaxReintentos(int maxReintentos) {
        this.maxReintentos = maxReintentos;
    }
    public boolean isEnvioCola() {
        return envioCola;
    }
    public void setEnvioCola(boolean envioCola) {
        this.envioCola = envioCola;
    }
}
```

El plugin necessita una configuració per funcionar d'una manera o un altre. A continuació s'expliquen cada un dels paràmetres de configuració.

- **ListadoDestinatariosModificable**: permet indicar si el llistat de destinataris es pot modificar.
- **ConfiguraciónAnexos**: especifica quina informació s'enviarà al segment d'Annex del registre d'entrada.

  1 = custodiaId + metadades + fitxer + firma. És a dir a dins el segment annexes de l'assentament s'enviaria tot el contingut de l'annexe.

  2 = custodiaId. A dins el segment annexes de l'assentament només s'enviaria l'Id del sistema que custodia l'arxiu.

  3 = custodiaId + metadades. A dins el segment annexes de l'assentament s'enviaria l'Id del sistema que custodia l'arxiu i les metadades del document.
- **MaxReintentos**: número de reintents que se volen per a distribuir el registre.
- **EnvioCola**: Indica si s'envia a la coa de distribució o no. A certs sistemes el temps de distribució pot ser molt llarg, per evitar tenir a l'usuari esperant a que finalitzi el procès, es pot enviar el registre a la coa de distribució i ja es distribueix de manera automàtica.

2.1.5.- Nova implementació

Se seguirà la estructura de la resta de plugins de distribució.

Aquesta estructura es pot consultar dins el repositori de regweb3.

https://github.com/GovernIB/registre/tree/registre-3.0/plugins/plugin-distribucion

2.1.6.- Configuració dins Regweb3

Per emprar la nova implementació que es faci al pom.xml del directori ear s'ha d'afegir la següent dependència:

```
 <dependency>
    <groupId>es.caib.regweb3</groupId>
    <artifactId>plugin-distribucion-novaimplementacio</artifactId>
    <version>${project.version}</version>
 </dependency>
```

2.2.- Plugin de Generació de Justificant

És el plugin que es configura per generar el justificant que s'associa a cada registre que es crea dins el registre. A continuació se detallen totes les accions que es poden fer a la generació de justificant.

L'API del plugin està formada per la següent classe i la interfície.

2.2.1.- Interfície IJustificantePlugin

```
public interface IJustificantePlugin extends IPlugin {

    public static final String JUSTIFICANTE_BASE_PROPERTY = IPLUGIN_BASE_PROPERTIES +
"postproceso.";

    /**
     * Metodo que genera el justificante de un registro de entrada.
     * @param registroEntrada registro de entrada del que se genera el justificante
     * @param url
     * @param specialValue
     * @param csv
     * @param idioma
     * @return
     * @throws Exception
     */
     byte[] generarJustificanteEntrada(RegistroEntrada registroEntrada, String url, String
                                       specialValue, String csv, String idioma) throws Exception;

    /**
     * Método que genera el justificante de un registro de entrada.
     * @param registroSalida registro de entrada del que se genera el justificante
     * @param url
     * @param specialValue
     * @param csv
     * @param idioma
     * @return
     * @throws Exception
     */
     byte[] generarJustificanteSalida(RegistroSalida registroSalida, String url, String
                                      specialValue, String csv, String idioma) throws Exception;

}
```

El mètode generarJustificanteEntrada ha de retornar un array de bytes del justificant generat pel registre d'entrada.

El mètode generarJustificanteSalida ha de retornar un array de bytes del justificant generat pel registre de sortida.

2.2.2.- Classe Justificante

```
public class Justificante {

    protected DocumentCustody justificant;
    protected List<Metadata> metadades;
    public DocumentCustody getJustificant() {
        return justificant;
    }
    public void setJustificant(DocumentCustody justificant) {
        this.justificant = justificant;
    }
    public List<Metadata> getMetadades() {
        return metadades;
```

```
    }
    public void setMetadades(List<Metadata> metadades) {
        this.metadades = metadades;
    }
}
```

Aquesta classe representa un justificant i està definit per un DocumentCustody(justificant) i una List<Metadata>(metadates).

2.2.3.- Nova implementació

Se seguirà l'estructura de la resta de plugins de justificant.

Aquesta estructura es pot consultar dins el repositori de regweb3.

https://github.com/GovernIB/registre/tree/registre-3.0/plugins/plugin-justificante

2.2.4.- Configuració dins Regweb3

Per a emprar la nova implementació que es faci al pom.xml del directori ear s'ha d'afegir la següent dependència:

```
<dependency>
    <groupId>es.caib.regweb3</groupId>
    <artifactId>plugin-justificante-novaimplementacio</artifactId>
    <version>${project.version}</version>
</dependency>
```

2.3.- Plugin de PostProcés

El plugin de post procés ens permet realitzar una serie de tasques a un sistema extern quan es crea un assentament, es crea un interessat a un assentament, etc. A continuació se detallen totes les accions que es pot fer a una tasca de postprocés.

2.3.1.- Interfície IPostProcesoPlugin

```
public interface IPostProcesoPlugin extends IPlugin {
    public static final String POSTPROCESO_BASE_PROPERTY = IPLUGIN_BASE_PROPERTIES + "postproceso.";
    /**
     * crear un registro nuevo
     * @param registroEntrada
     * @return
     * @throws Exception
     */
    public void nuevoRegistroEntrada(RegistroEntrada registroEntrada) throws Exception;
    /**
     * crear un registro nuevo
     * @param registroSalida
     * @return
     * @throws Exception
     */
    public void nuevoRegistroSalida(RegistroSalida registroSalida) throws Exception;
    /**
     * actualizar un registro
     * @param registroEntrada
     * @return
     * @throws Exception
     */
    public void actualizarRegistroEntrada(RegistroEntrada registroEntrada) throws Exception;
    /**
     * actualizar un registro
     * @param registroSalida
     * @return
     * @throws Exception
     */
    public void actualizarRegistroSalida(RegistroSalida registroSalida) throws Exception;
    /**
     * nuevo interesado
     * @param interesado
     * @param numeroEntrada
     * @return
     * @throws Exception
     */
    public void nuevoInteresadoEntrada(Interesado interesado, String numeroEntrada) throws
Exception;
    /**
     * nuevo interesado
     * @param interesado
     * @param numeroSalida
     * @return
     * @throws Exception
     */
    public void nuevoInteresadoSalida(Interesado interesado, String numeroSalida) throws Exception;
    /**
     * actualizar interesado
     * @param interesado
     * @param numeroEntrada
     * @return
     * @throws Exception
     */
    public void actualizarInteresadoEntrada(Interesado interesado, String numeroEntrada) throws
Exception;
    /**
     * actualizar interesado
```

```
     * @param interesado
     * @param numeroSalida
     * @return
     * @throws Exception
     */
    public void actualizarInteresadoSalida(Interesado interesado, String numeroSalida) throws
Exception;
    /**
     * eliminar interesado
     * @param idInteresado
     * @param numeroEntrada
     * @return
     * @throws Exception
     */
    public void eliminarInteresadoEntrada(Long idInteresado, String numeroEntrada) throws Exception;
    /**
     * eliminar interesado
     * @param idInteresado
     * @param numeroSalida
     * @return
     * @throws Exception
     */
    public void eliminarInteresadoSalida(Long idInteresado, String numeroSalida) throws Exception;
```

En aquest cas, s'haurà d'implementar la funcionalitat que es vulgui fer al sistema extern per cada un dels mètodes especificats.

2.3.2.- Nova implementació

Se seguirà l'estructura de la resta de plugins de postprocés.

Aquesta estructura es pot consultar dins el repositori de regweb3.

https://github.com/GovernIB/registre/tree/registre-3.0/plugins/plugin-postproceso

2.3.3.- Configuració dins Regweb3

Per a emprar la nova implementació que es faci al pom.xml del directori ear s'ha d'afegir la següent dependència:

```
<dependency>
    <groupId>es.caib.regweb3</groupId>
    <artifactId>plugin-postproceso-novaimplementacio</artifactId>
    <version>${project.version}</version>
</dependency>
```

3.- Plugins externs a Regweb3

Tota la informació referent als plugins externs que empra Regweb3 la podeu trobar als enllaços indicats. A més se fa una petita explicació de com s'integra regweb3 amb aquests plugins.

3.1.- Plugin User Information

https://github.com/GovernIB/pluginsib-userinformation/tree/pluginsib-userinformation-2.0

En aquest cas la cridada al plugin des de regweb3 es fa a partir de l'identificador d'un usuari de la següent manera:

```
IUserInformationPlugin loginPlugin = (IUserInformationPlugin) pluginEjb.getPlugin(null,
RegwebConstantes.PLUGIN_USER_INFORMATION)
UserInfo regwebUserInfo = loginPlugin.getUserInfoByUserName(identificador);
```

3.2.- Plugin Scan Web

Podreu trobar tota la documentació aquí:

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-2.0

Des de la versió de registre 3.2 la versió que s'empra és

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-4.0

3.2.1.- Plugin Scan Web-DIGITALIB

Podreu trobar tota la documentació aquí:

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-2.0/digitalib-plugin

Nota: s'han de definir les següents propietats de metadades per passar-li al plugin amb aquests valors en concret:

```
es.caib.regweb3.pluginsib.scanweb.digitalib.metadata.functionary.username=functionary.username
es.caib.regweb3.pluginsib.scanweb.digitalib.metadata.functionary.fullname=functionary.fullname
es.caib.regweb3.pluginsib.scanweb.digitalib.metadata.functionary.administrationid=functionary.administrationid
es.caib.regweb3.pluginsib.scanweb.digitalib.metadata.document.language=document.language
```

Des de la versió de registre 3.2 la versió que s'empra és

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-4.0/digitalib-plugin

Des d'aquesta versió ja no és necessari indicar les variables de les metadades

Des de la versió 3.2.1 és obligatori definir la següent propietat global a nivell aplicació(administrador)

```
es.caib.regweb3.scanweb.absoluteurl = http://dominiregweb/regweb3
```

Exemple:

```
es.caib.regweb3.scanweb.absoluteurl = http://registre3.fundaciobit.org/regweb3
```

Per incloure aquest perfil a l'aplicació s'ha de compilar de la següent forma:

```
mvn -DskipTests clean install -Pdigitalibscanweb
```

3.2.2.- Iecisa

Podreu trobar tota la documentació aquí:

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-2.0/iecisa-scanweb-plugin

Des de la versió registre-3.2 la versió del plugin que s'empra és:

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-4.0/iecisa-scanweb-plugin

Per incloure aquest perfil a l'aplicació s'ha de compilar de la següent forma:

```
mvn -DskipTests clean install -Piecisascanweb
```

En el cas del plugin d'scan web regweb3 s'integra amb l'api a través del Bean «ScanWebModuleEjb.java» que és l'encarregat de preparar tot i fer totes les cridades necessaries.

https://github.com/GovernIB/registre/blob/registre-3.1/persistence/src/main/java/es/caib/regweb3/persistence/ejb/ScanWebModuleEjb.java

Des de la versió registre-3.2 el Bean s'ha hagut d'adaptar

https://github.com/GovernIB/registre/blob/registre-3.2/persistence/src/main/java/es/caib/regweb3/persistence/ejb/ScanWebModuleEjb.java

3.2.3.- Plugin Scan Web - Digital Massiu

Aquest plugin s'empra a partir de la versió de registre-3.2

https://github.com/GovernIB/pluginsib-scanweb/tree/pluginsib-scanweb-4.0/digitalibmassive-plugin

Per incloure aquest perfil a l'aplicació s'ha de compilar de la següent forma:

```
mvn -DskipTests clean install -Pdigitalibmassivescanweb
```

3.3.- Plugin Signatura Servidor

https://github.com/GovernIB/portafib/tree/portafib-2.0/pluginsib-signatureserver

Regweb3 s'integra amb aquest plugin a través del Bean «SignatureServerBean.java». La implementació es pot veure aquí:

https://github.com/GovernIB/registre/blob/registre-3.1/persistence/src/main/java/es/caib/regweb3/persistence/ejb/SignatureServerBean.java

A continuació es mostra esquemàticament els mètodes que intervenen en aquesta integració:

checkDocumentAndSignature

firmaCAdESEPESDetached

firmaPAdESEPES

signFile

signJustificante

```
public I18NTranslation checkDocumentAndSignature(AnexoFull input, long idEntidad,
                                                 boolean sir, Locale locale, boolean
force, String numeroRegistro) throws I18NException
{
...
firmaCAdESEPESDetached(input, doc, locale, signaturePlugin, idEntidad, numeroRegistro);


...
}

public void firmaPAdESEPES(AnexoFull input, long idEntidad, Locale locale, String
numeroRegistro) throws I18NException{
...

SignatureCustody sc = signFile(docToSign, signType, signMode, epes,
        signaturePlugin, locale, reason, idEntidad, new Date(), new StringBuilder(),
numeroRegistro);

...
}
```

3.4.- Plugin Custòdia

En aquest cas hi ha múltiples implementacions d'aquest plugin i es poden trobar totes aquí

https://github.com/GovernIB/pluginsib-documentcustody/tree/pluginsib-documentcustody-2.0

Si consultam el manual de l'api «ManualProgramadorApiDocumentCustody.pdf» en concret la pg 8 veim que la cridada a custodia es fa de la següent manera:

```
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);
```

Regweb3 empra dues implementacions d'aquest plugin «FileSystem» i «ArxiuCaib», però a les dues implementacions la cridada específica des de regweb3 es fa així:

```
protected Map<String, Object> getCustodyParameters(IRegistro registro, Anexo anexo,
                                                   AnexoFull anexoFull, UsuarioEntidad
usuarioEntidad) {

    Map<String, Object> custodyParameters = new HashMap<String, Object>();

    custodyParameters.put("registro", registro);
    custodyParameters.put("anexo", anexo);
    custodyParameters.put("anexoFull", anexoFull);
    custodyParameters.put("usuarioEntidad", usuarioEntidad);

    Interesado interesado = registro.getRegistroDetalle().getInteresados().get(0);
    custodyParameters.put("ciudadano_nombre", interesado.getNombreCompleto());

    String documentAdministratiu = interesado.getDocumento();
    if (documentAdministratiu == null) {
        documentAdministratiu = interesado.getCodigoDir3();
    }

    custodyParameters.put("ciudadano_idadministrativo", documentAdministratiu);

    return custodyParameters;
}

String custodyID = documentCustodyPlugin.reserveCustodyID(custodyParameters);
```

3.5.- Plugin Informació/Validació Firmes

https://github.com/GovernIB/pluginsib-validatesignature/tree/pluginsib-validatesignature-2.0

En aquest cas Regweb3 s'integra amb aquest plugin a través d'aquests 2 mètodes del SignatureServerBean.java

A continuació es mostra esquemàticament aquesta integració, però es pot consultar tot el codi aquí:

https://github.com/GovernIB/registre/blob/registre-3.1/persistence/src/main/java/es/caib/regweb3/persistence/ejb/SignatureServerBean.java

checkDocument

```
public I18NTranslation checkDocument(AnexoFull input, long idEntidad, Locale locale,
boolean force) throws I18NException{

...
ValidateSignatureResponse resp = new ValidateSignatureResponse();
resp = callToValidaFirma(locale, sign, doc,idEntidad);
....
}

protected ValidateSignatureResponse callToValidaFirma(Locale locale, SignatureCustody
sign,DocumentCustody doc, Long idEntidad) throws I18NException{
...
resp = validatePlugin.validateSignature(validationRequest);
...
}
```