



Manual de *WebServices* i CallBack de PortaFIB

**Manual de Integració dels Serveis Web disponibles a
PortaFIB i les diferents formes de Gestionar els
CallBack.**



G VICEPRESIDÈNCIA
O I CONSELLERIA
I INNOVACIÓ,
B RECERCA I TURISME
/ DIRECCIÓ GENERAL
DESENVOLUPAMENT
TECNOLÒGIC



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

Informació general del document.

Descripció.

Títol:	Manual de WebServices de PortaFIB
Estat:	Esborrany/Aprovat
Versió:	1.0
Autor/s:	Antoni Nadal Bennasar
Creat:	14/01/2014
Modificat	26/01/2016
Fitxer:	Manual_de_WebServices_de_PortaFIB.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
Jar compilat del repositori maven de sourceforge	A. Nadal	12/09/2014
Client requereix wsdl	A. Nadal	23/09/2014
Nou mètode per borrar plantilles i per obtenir usuari-entitat a partir de l'identificador d'usuari-persona (o username)	A. Nadal	19/11/2014
Més informació sobre Callback	A. Nadal	26/01/2016
Callback REST	A. Nadal	18/08/2017

Font documental.

Index de Contingut

1.-Introducció.....	6
1.1.-Introducció General.....	6
1.2.-Rols dels Usuaris Aplicació.....	7
1.3.-API de Comunicació.....	7
1.3.1.-Generar codi a partir de WSDL.....	7
1.3.2.-Emprar llibreries compilades.....	8
1.3.3.-Utilitzar jar de l'API del repositori maven.....	8
1.4.-Instanciar API de WebService.....	9
1.5.-Mètodes Comuns a Totes les APIs.....	10
1.5.1.-String getVersion() i int getVersionWs().....	10
1.5.2.-String getEntitatID().....	10
1.5.3.-List<String> getSupportedLanguages().....	11
1.5.4.-FitxerBean downloadFileUsingEncryptedFileID(encryptedFileID).....	11
1.6.-Gestió d'Errors	13
1.6.1.-Exception.....	13
1.6.2.-Wsl18NException.....	14
1.6.3.-WsValidationException.....	15
1.7.-Constants PortaFIB.....	17
1.8.-Traduccions PortaFIB.....	17
2.-API d'Usuaris-Aplicació.....	17
2.1.-Introducció.....	17
2.2.-Mètodes de l'API d'Usuari-Aplicació.....	18
2.2.1.-createUsuariAplicacio().....	18
2.2.2.-getUsuariAplicacio().....	18
2.2.3.-addRolUserToUsuariAplicacio().....	19
2.2.4.-removeRolUserToUsuariAplicacio().....	19
2.2.5.-addRolAdminToUsuariAplicacio().....	20
2.2.6.-removeRolAdminToUsuariAplicacio().....	21
2.2.7.-deleteUsuariAplicacio().....	21
2.2.8.-listUsuariAplicacio().....	22
3.-API Usuaris-Persona, Usuaris-Entitat i Càrrecs.....	23
3.1.-Mètodes Usuari Persona.....	24
3.1.1.-getInfoFromPluginUserInfoByAdministrationID().....	24
3.1.2.-getInfoFromPluginUserInfoByUsername().....	24
3.1.3.-createUsuariPersona().....	24
3.1.4.-getUsuariPersona().....	25
3.1.5.-getUsuariPersonalIDByAdministrationID().....	25
3.1.6.-deleteUsuariPersona().....	26
3.2.-Mètodes Usuari Entitat.....	27
3.2.1.-createUsuariEntitat().....	27
3.2.2.-CreateUsuariEntitatSimple().....	27
3.2.3.-getUsuariEntitat().....	28
3.2.4.-getUsuariEntitatIDInMyEntitatByAdministrationID().....	28
3.2.5.-getUsuariEntitatIDByAdministrationID().....	29
3.2.6.-getUsuariEntitatIDInMyEntitatByUsuariPersonalID().....	29
3.2.7.-getUsuariEntitatIDByUsuariPersonalID().....	30
3.2.8.-activateUsuariEntitat().....	30

3.2.9.-deactivateUsuariEntitat()	31
3.2.10.-addRoleAdministradorDeEntitat()	31
3.2.11.-removeRoleAdministradorDeEntitat()	31
3.2.12.-addRoleSolicitant()	32
3.2.13.-removeRoleSolicitant()	32
3.2.14.-deleteUsuariEntitat()	32
3.3.-Mètodes Càrrec	33
3.3.1.-createCarrecSimple()	33
3.3.2.-createCarrec()	34
3.3.3.-getCarrec()	34
3.3.4.-getCarrecsByEntitatID()	34
3.3.5.-getCarrecsOfMyEntitat()	35
3.3.6.-updateAdministrationIDOfCarrec()	35
3.3.7.-activateCarrec()	36
3.3.8.-deactivateCarrec()	36
3.3.9.-deleteCarrec()	36
4.-API Petició de Firma	37
4.1.-Introducció	37
4.2.-Utilitats de l'API: classe PeticioDeFirmaUtils	37
4.3.-Mètodes de l'API PortaFIBPeticioDeFirma	38
4.3.1.-createPlantillaFluxDeFirmes()	39
4.3.2.-deletePlantillaFluxDeFirmes()	40
4.3.3.-instantiatePlantillaFluxDeFirmes()	40
4.3.4.-getCurrentCustodiaPluginClass()	40
4.3.5.-getDefaultCustodiaInfo()	41
4.3.6.-getTipusDeDocuments()	41
4.3.7.-createPeticioDeFirma()	42
4.3.8.-startPeticioDeFirma()	42
4.3.9.-createAndStartPeticioDeFirma()	43
4.3.10.-getPeticioDeFirma()	43
4.3.11.-getLastSignedFileOfPeticioDeFirma()	44
4.3.12.-getStateOfPeticioDeFirma()	44
4.3.13.-pausePeticioDeFirma()	45
4.3.14.-resetPeticioDeFirma()	45
4.3.15.-deletePeticioDeFirma()	46
4.4.-Exemples	46
5.-API Callback WS PortaFIB v1	49
5.1.-Introducció	49
5.2.-Crear WebService	50
5.3.-API de Callback	53
5.4.-Classes WS Callback	54
5.4.1.-Classe PortalFIBEvent	54
5.4.2.-Classe SigningRequest	55
5.4.3.-Classe Sign	55
5.4.4.-Classe Actor	55
6.-Callback REST PortaFIB v1	56
6.1.-Introducció	56
6.2.-Implementar servei REST en aplicació externa	56
6.2.1.-Dependències	56
6.2.2.-Controlador de Servei REST	57
6.2.3.-Modificacions en el web.xml	58
7.-Annexes	59

7.1.-Classe UsuariAplicacioBean.....	59
7.2.-Classe UsuariAplicacioFilterWs.....	60
7.3.-Classe UsuariPersonaBean.....	61
7.4.-Classe UsuariEntitatBean.....	61
7.5.-Classe CarrecWs.....	62
7.6.-Classe FluxDeFirmesWs.....	62
7.7.-Classe BlocDeFirmesWs.....	63
7.8.-Classe FirmaBean.....	63
7.9.-Classe CustodiaInfoBean.....	64
7.10.-Classe TipusDocumentInfoWs.....	67
7.11.-Classe MetadadaBean.....	67
7.12.-Classe AnnexBean.....	68
7.13.-Classe PeticioDeFirmaWs.....	69

1.- Introducció

1.1.- Introducció General

Abans d'entrar en en l'explicació dels WebServices és molt convenient llegir la introducció del document Manual d'Usuari de PortaFIB ja que conté informació general i específica del funcionament intern de PortaFIB que ens ajudarà a entendre millor els WebServices.

Els webservices estan agrupats per categories, per la qual cosa també seria interessant revisar l'apartat concret en el Manual d'Usuari de PortaFIB per la mateixa raó descrita en el paràgraf anterior, per exemple, pel webservice d'Usuaris Aplicació revisar els punts de Gestió d'Usuaris Aplicació de l'Administrador i de l'Administrador d'Entitat del el Manual d'Usuari de PortaFIB.

Les categories de les que s'ha parlat anteriorment són:

Nom	Descripció	Context
HelloWorld	API que es pot cridar sense autenticar-se i serveix per provar la comunicació amb el servidor. No requereix autenticació.	/portafib/ws/v1/PortaFIBHelloWorld
Usuari Aplicació	Gestiona els Usuaris Aplicació. Requereix autenticació.	/portafib/ws/v1/PortaFIBUsuariAplicacio
Usuari Entitat	Gestiona els Usuaris Persona, els Usuaris Entitat i els Càrrecs. Requereix autenticació.	/portafib/ws/v1/PortaFIBUsuariEntitat
Petició de Firma	Gestiona les Peticions de Firma, Custòdia i Plantilles de Flux de Firms	/portafib/ws/v1/PortaFIBPeticioDeFirma

Per exemple, en un servidor en el nostre ordinador, la ruta per accedir als serveis de HelloWorld seria la següent <http://localhost:8080/portafib/ws/v1/PortaFIBHelloWorld> i per accedir al WSDL de HelloWorld seria la següent <http://localhost:8080/portafib/ws/v1/PortaFIBHelloWorld?wsdl>.

1.2.- Rols dels Usuaris Aplicació

Els usuaris aplicació són els usuaris autoritzats a connectar-se via webservices amb el PortaFIB. Per permetre nivells d'accés a diferents mètodes, cada usuari aplicació pot tenir dos rols associats:

- **ROLE USUARI (PFI_USER):** Estaria associat als rols virtuals Sol·licitant, Destinatari, Delegat i Col·laborador que tenim a la capa web i la tasca principal seria la de poder crear Peticions de Firma.
- **ROLE ADMIN (PFI_ADMIN):** Estaria associat amb els rols Administrador i Administrador d'Entitat de la capa web, i serviria per realitzar totes les tasques administratives (a nivell de PortaFIB o a nivell d'entitat. Recordar que PortaFIB és multientitat, i que un usuari aplicació requereix que estigui assignat a una entitat en concret, però que amb aquest rol també podrà realitzar tasques d'Administració que afecten a PortaFIB a part de les tasques pròpies d'administració de l'entitat.

També un usuari aplicació no pot tenir cap rol associat per la qual cosa es trobaria en un estat similar al de desactivació però només al que es refereix a cridades webservices.

Quan es detalli cada mètode de webservice s'indicarà quins rols es requereixen per poder cridar al mètode en qüestió.

Com és lògic, els webservices no autenticats no requereixen d'usuari aplicació per la cridada dels seus mètodes cosa que fa sense sentit els rols descrits en aquest punt.

1.3.- API de Comunicació

Per connectar-se amb el servidor es necessita un conjunt de classes java (o qualsevol altra llenguatge) que anomenarem API de comunicació. Aquesta API de comunicació està formada per elements Bean que contindran la informació a passar i uns gestors que són classes amb mètodes que són les accions disponibles per cada webservice. El codi necessari per connectar-se al servidor via webservices es pot aconseguir de dues formes:

- Generar codi a partir de wsdl o
- Emprar llibreries compilades
- Utilitzar jar de l'API del repositori maven

Es recomana la tercera opció ja que aquesta opció inclou classes d'utilitat de client webservice com classes amb les constants del projecte PortaFIB, mètodes d'utilitat de gestió d'excepcions i fitxers de traduccions, i a més no s'ha d'anar a cercar el jar per la web de sourceforge.

1.3.1.- Generar codi a partir de WSDL

Per exemple per obtenir l'API de comunicació del webservice HelloWorld executaríem la següent comanda:

```
wsconsume -k http://localhost:8080/portafib/ws/v1/PortaFIBHelloWorld?wsdl -s src/main/java -n -p es.caib.portafib.ws.api.v1
```

El programa wsconsume és un script (.bat i .sh) que es troba en el directori bin del JBoss al que abans se li ha instal·lat el *patch* de CXF i el que fa és generar el codi java necessari per connectar-se al webservice.

1.3.2.- Emprar llibreries compilades

Com ja s'ha comentat la utilització de les APIs ja compilades són la segona millor opció ja que inclou un conjunt de classes d'utilitat com són les constants utilitzades en les cridades a l'API, la gestió d'errors i la gestió de traduccions (així com els fitxers de traduccions de PortaFIB)

Aquesta API la podem trobar a la següent ruta `ws/portafib_api/target/portafib-ws-api-1.0.0.jar` si compilam el codi de PortaFIB o en la mateixa ruta si ens descarregam el zip del binari de PortaFIB que trobarem dins l'apartat de *files* de <http://portafib.sourceforge.net>.

Algunes de les classes addicionals que inclou aquest jar:

- **es.caib.portafib.ws.api.v1.utils.WsClientUtils**: Conté mètodes per transformar excepcions del tipus `Wsl18NException` i `WsValidationException` a cadena de caràcters. Veure punt “1.6.-Gestió d'Errors”
- **es.caib.portafib.ws.api.v1.utils.PeticioDeFirmaUtils**: Conté mètodes per crear peticions de forma simple i ràpida a partir d'un mínim de dades i un conjunt de nifs. També crea Annexes a partir de fitxers i flux de firmes a partir de llistes de NIFs.
- **es.caib.portafib.ws.api.v1.utils.l18NUtils**: utilitza els fitxers de traduccions de PortaFIB per traduir els codis de missatge de les excepcions a algun idioma determinat. Veure punt “1.8.-Traduccions PortaFIB”
- **es.caib.portafib.utils.Constants**: Classe que conté totes les constants utilitzades per PortaFIB i que ajuden en la tasca de desenvolupar amb l'API. Veure punt “1.7.-Constants PortaFIB”.

1.3.3.- Utilitzar jar de l'API del repositori maven

És el mateix jar descrit en el punt anterior però no l'hem d'anar a cercar a cap pàgina web sinó que està emmagatzemat dins d'un repositori maven que s'ha fet dins el projecte PortaFIB de sourceforge. Només hem d'afegir el repositori de PortaFIB dins el nostre pom.xml del projecte:

```
<repositories>
<repository>
  <id>es.caib.portafib</id>
  <name>PortaFIB Maven Repository</name>
  <url>http://portafib.sourceforge.net/maven/</url>
</repository>
</repositories>
```

I a més afegir la dependència al jar corresponent:


```
<dependencies>
  <dependency>
    <artifactId>portafib-ws-api</artifactId>
    <groupId>es.caib.portafib</groupId>
    <version>${project.version}</version>
  </dependency>
</dependencies>
```

1.4.- Instanciar API de WebService

El serveis descrits en el punt 1.1.-Introducció General és poden instanciar com es mostra a continuació. Només requerim un projecte amb un jar que inclogui l'api del punt 1.3.-API de Comunicació.

```
String endpoint = "http://localhost:8080/portafib/ws/v1/PortaFIBHelloWorld";
URL wsdl = new URL(endpoint + "?wsdl");

PortaFIBHelloWorldWsService helloService;
helloService = new PortaFIBHelloWorldWsService(wsdl);
PortaFIBHelloWorldWs hello = helloService.getPortaFIBHelloWorldWs();

// Adreça servidor
Map<String, Object> reqContext = ((BindingProvider) hello).getRequestContext();
reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, endpoint);
```

L'API anterior no necessita autenticació, però la resta d'API si. A continuació es mostra com instanciar la resta d'API indicant l'usuari i contrasenya de l'usuari aplicació:

```
String endpoint="http://localhost:8080/portafib/ws/v1/PortaFIBPeticioDeFirma";
URL wsdl = new URL(endpoint + "?wsdl");

PortaFIBPeticioDeFirmaWsService service;
service = new PortaFIBPeticioDeFirmaWsService(wsdl);
PortaFIBPeticioDeFirmaWs api = service.getPortaFIBPeticioDeFirmaWs();

// Adreça servidor i credencials
String usr_app = "user_app"; // username application
String pwd_app = "mypassword"; // password application
Map<String, Object> reqContext = ((BindingProvider) api).getRequestContext();
reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, endpoint);
reqContext.put(BindingProvider.USERNAME_PROPERTY, usr_app);
reqContext.put(BindingProvider.PASSWORD_PROPERTY, pwd_app);

System.out.println("Versio " + api.getVersion());
```

Les dues línies en negreta introdueixen dins la petició l'usuari i contrasenya de l'usuari aplicació que volem utilitzar per fer la cridada.

1.5.- Mètodes Comuns a Totes les APIs

Cada WebService conté uns mètodes d'utilitat comuns a totes les APIs i que són:

1.5.1.- String getVersion() i int getVersionWs()

Retorna la versió del PortaFIB i la versió dels WebService del PortaFIB respectivament.

Vegem un exemple emprant l'API instanciada en el punt anterior:

```
System.out.println("Versió PortaFIB : " + hello.getVersion());  
System.out.println("Versió PortaFIB-WS: " + hello.getVersionWs());
```

I la sortida de l'execució del programa obtindríem una cosa com:

```
Versió PortaFIB : 1.0.0  
Versió PortaFIB-WS: 1
```



En l'entorn de la CAIB (Govern Balear) la versió de PortaFIB sempre inclourà un sufix “-caib” de forma que l'execució del programa anterior en un servidor PortaFIB de la CAIB mostrarà per pantalla el següent:

```
Versió PortaFIB : 1.0.0-caib  
Versió PortaFIB-WS: 1
```

1.5.2.- String getEntitatID()

Retorna l'identificador d'entitat al que està associat l'usuari aplicació que s'ha emprat en la comunicació per autenticar-se. Recordar que aquest mètode només estarà disponible en les comunicacions autenticades.

Per mostrar un exemple, farem ús de l'API de gestió dels Usuaris Aplicació a que haurem d'indicar l'usuari i la contrasenya de l'usuari aplicació amb que es realitzarà l'autenticació en el servidor:

```
final String username = "fundaciobit_usrapp";
final String password = "fundaciobit_usrapp";
final String endpoint = "http://localhost:8080/portafib/ws/v1/PortaFIBUsuariAplicacio";
URL wsdl = new URL(endpoint + "?wsdl");

PortaFIBUsuariAplicacioWsService service = new PortaFIBUsuariAplicacioWsService(wsdl);
PortaFIBUsuariAplicacioWs api = service.getPortaFIBUsuariAplicacioWs();

// Adreça, Usuari i Contrasenya
Map<String, Object> reqContext = ((BindingProvider) usuarios).getRequestContext();
reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, endpoint);
reqContext.put(BindingProvider.USERNAME_PROPERTY, username);
reqContext.put(BindingProvider.PASSWORD_PROPERTY, password);

System.out.println("L'usuari aplicació " + username
    + " pertany a l'entitat amb ID " + api.getEntitatID());
```

La sortida de l'execució d'aquest programa seria una cosa com:

```
L'usuari aplicació fundaciobit_usrapp pertany a l'entitat amb ID fundaciobit
```

1.5.3.- List<String> getSupportedLanguages()

Retorna una llista del idiomes suportats per PortaFIB en format “ISO 639-1”.

Per exemple, després de l'execució del següent tros de codi java:

```
...
System.out.println("Idiomes:" +
    Arrays.toString(api.getSupportedLanguages().toArray()));
...
```

Per la consola ens apareixeria una cosa com:

```
Idiomes:[ca, es]
```

1.5.4.- FitxerBean downloadFileUsingEncryptedFileID(encryptedFileID)

1.5.4.1.- Classe FitxerBean

Hi ha alguns elements de l'API que requereixen fer referències a fitxers. Aquesta referència ve representada per un altre Bean anomenat FitxerBean que es una classe java que conté els següents camps:

Camp	Tipus	Descripció
fitxerID	long	Identificador del fitxer. No es requereix durant la creació.
nom	String	Nom del fitxer
descripcio	String	(Opcional) Descripció del que conté el fitxer
tamany	long	Tamany del fitxer en bytes
mime	String	MIME del que conté el fitxer (text/plain, application/pdf, ...)
data	byte[]	(Opcional) El contingut del fitxer. El camp data només s'utilitza durant les peticions webservice (request) ja que pels elements de retorn aquest camp val null excepte pels mètodes que específicament es desenvolupin per descarregar fitxers.
encryptedFileID	String	(Opcional) Hash utilitzat per recuperar el contingut del fitxer. Per recuperar el contingut del fitxer no es pot utilitzar fitxerID ja que alguna persona podria recuperar documents d'altres peticions fent un simple recorregut dels valors possibles de fitxerID.

Tal i com es comenta en el camp data, aquest sempre estarà buit excepte quan es realitzin peticions webservices cap el servidor (enviar fitxers) o en mètodes específicament desenvolupats per descarregar fitxers.

Un exemple d'aquest darrer cas és el mètode `downloadFileUsingEncryptedFileID()` que descarrega la informació d'un fitxer i el seu contingut i que es troba en totes les APIs en que es requereix autenticació:

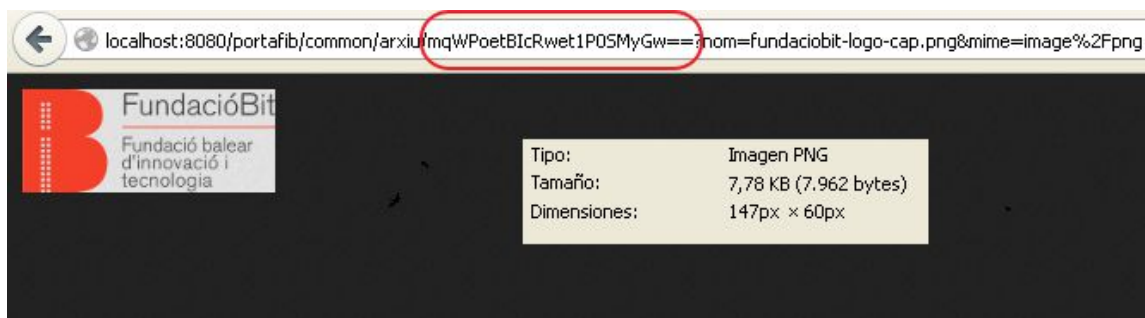
1.5.4.2.- Prototipus

```
FitxerBean downloadFileUsingEncryptedFileID(String encryptedFileID);
```

L'identificador "encryptedFileID" l'obtidrem del camp encryptedFileID dels elements FitxerBean del objectes retornats des del servidor.



Comentar que l'entorn web també empra el mateix sistema de hash o encryptedFileID, per la qual cosa es podem emprar imatges i arxius per fer proves amb aquest mètode. Aquí mostrem la URL del logo de l'entitat i on podem extreure l'encryptedFileID:



El següent programa descarrega aquesta imatge i mostra informació de la mateixa, que com veurem concorda amb la informació del fitxer que ens indica el navegador:

```
String encryptedFileID = "mqWPoetBicRwet1P0SMYgw=="; // Logo FundacioBIT
FitxerBean fitxer = usuaris.downloadFileUsingEncryptedFileID(encryptedFileID);

System.out.println(" ID: " + fitxer.getFitxerID());
System.out.println(" ENCODED ID: " + fitxer.getEncryptedFileID());
System.out.println(" LEN BBDD: " + fitxer.getTamany() + " bytes");
System.out.println(" DATA LEN: " + fitxer.getData().length + " bytes");
System.out.println(" MIME: " + fitxer.getMime());
```

La sortida per consola de l'execució seria una cosa com:

```
ID: 80151
ENCODED ID: mqWPoetBicRwet1P0SMYgw==
LEN BBDD: 7962 bytes
DATA LEN: 7962 bytes
MIME: image/png
```

1.6.- Gestió d'Errors

Les excepcions controlades que llança PortaFib estan preparades per ser traduïdes en el client, encara que també s'envia el missatge ja traduït amb l'idioma que es va definir quan es va donar d'alta l'usuari aplicació.

Existeixen tres tipus d'excepcions:

1.6.1.- Exception

Són excepcions no controlades per PortaFIB o gestionades pel propi sistema de webservices (camps o paràmetres requerits, camps o paràmetres no null, autenticació, autorització, ...). Amb un simple try-catch es poden capturar.

A causa de la diversitat de tipus d'excepcions s'engloben en Exception.

Aquí és mostra el try-catch per l'exemple del punt 1.5.2.-String getEntitatID():

```
try {
    UsuariAplicacioBean usuariAplicacioBean = new UsuariAplicacioBean();
    usuaris.createUsuariAplicacio(usuariAplicacioBean);
} catch (Exception e) {
    System.err.println("Exception: " + e.getMessage());
    e.printStackTrace(System.err);
}
```

Si no existeix el servidor o no està en marxa:

```
Exception: HTTP transport error: java.net.UnknownHostException: localhost22
com.sun.xml.internal.ws.client.ClientTransportException: HTTP transport error:
java.net.UnknownHostException: localhost22
```

Si l'usuari no té rols, l'usuari no és correcte o la contrasenya no és correcte llavors l'error és:

```
Exception: The server sent HTTP status code 401: No Autorizado
com.sun.xml.internal.ws.client.ClientTransportException: The server sent HTTP
status code 401: No Autorizado
```

Si l'usuari no té els rols requerits pel mètode o servei que crida llavors l'error és:

```
Exception: Caller unauthorized
javax.xml.ws.soap.SOAPFaultException: Caller unauthorized
```

Quan algun paràmetre es valida en la capa de webservices (En aquest cas s'ha cridat al mètode createUsuariAplicacio() del servei PortaFIBUsuariAplicacioWs):

```
Exception: Method Parameter: usuariAplicacioBean cannot be null. This is BP 1.1
R2211 violation.
javax.xml.ws.WebServiceException: Method Parameter: usuariAplicacioBean cannot
be null. This is BP 1.1 R2211 violation.
```

1.6.2.- WsI18NException

Aquesta excepció serveix per indicar algun error a l'hora de realitzar alguna acció o validacions simples en els paràmetres dels mètodes.

En el programa següent intentam donar d'alta un Usuari-Entitat i associar-ho a la Persona amb nif "NIF_inexistent" cosa que fallarà irremediament ja que aquest NIF a part de no existir no té un format correcte:

```
try {
    String endpoint = "http://localhost:8080/portafib/ws/v1/PortaFIBUsuariEntitat";

    PortaFIBUsuariEntitatWsService service = new PortaFIBUsuariEntitatWsService();

    PortaFIBUsuariEntitatWs usuarios = service.getPortaFIBUsuariEntitatWs();

    // Adreça, Usuari i contrasenya
    Map<String, Object> reqContext = ((BindingProvider) usuarios).getRequestContext();
    reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, endpoint);
    reqContext.put(BindingProvider.USERNAME_PROPERTY, "fundaciobit_usrapp");
    reqContext.put(BindingProvider.PASSWORD_PROPERTY, "fundaciobit_usrapp");

    usuarios.createUsuariEntitatSimple("NIF_inexistent", "fundaciobit");
} catch (WsI18NException i18n) {
    System.err.println(" ===== I18NException =====");
    System.err.println(WsClientUtils.toString(i18n));
} catch (Exception e) {
    System.err.println("Exception: " + e.getMessage());
    e.printStackTrace(System.err);
}
```

La classe WsClientUtils (paquet es.caib.portafib.ws.api.v1.utils.WsClientUtils) conté algunes rutines per imprimir excepcions.

L'estructura d'una excepció WsI18NException és la següent:

Camp	Tipus	Descripció
message	String	Conté el missatge d'error compost i traduït en l'idioma definit en la informació de l'usuari aplicació que realitza la cridada.
faultInfo	Wsl18NError	Classe que encapsula la informació de traducció
translation	Wsl18NTranslation	Conté els codis i paràmetres per traduir manualment aquest error
code	String	Conté el codi de traducció del missatge
args	Wsl18NArgument[]	Arguments del missatge
translate	boolean	Indica si aquest argument s'ha de traduir o no
value	String	Conté un valor final per cert argument o el codi que s'ha de traduir.

La sortida per la cridada del programa que hem vist abans seria la següent:

```
===== I18NException =====
MESSAGE: No existeix usuari persona amb NIF NIF_inexistent
TRANSLA: usuaripersona.noexisteix [*nif, NIF_inexistent]
```

1.6.3.- WsValidationException

Aquesta excepció és llançada si hi ha algun error o alguns errors de validació en algun o alguns camps de cert Bean. Normalment és llançada en els mètodes de creació o actualització d'estructures de dades.

En el següent exemple volem crear un Usuari Entitat emprant el bean UsuariEntitatBean, al que no establim cap valor en cap dels seus camps.

```
try {

    Holder<UsuariEntitatBean> usuariEntitatHolder;
    usuariEntitatHolder = new Holder<UsuariEntitatBean>();
    usuariEntitatHolder.value = new UsuariEntitatBean();
    usuarios.createUsuariEntitat(usuariEntitatHolder);

} catch (WsValidationException ve) {
    System.err.println(" ===== ValidationException =====");
    System.err.println(WsClientUtils.toString(ve));
} catch (Exception e) {
    System.err.println("Exception: " + e.getMessage());
    e.printStackTrace(System.err);
}
```

En aquest cas també utilitzam la classe WsClientUtils i un mètode per imprimir l'excepció. La sortida de l'execució del programa és la següent:

```

===== Validation_Exception =====
ID(usuariEntitatID): El camp ID és obligatori.
Persona(usuariPersonaID): El camp Persona és obligatori.
Entitat(entitatID): El camp Entitat és obligatori.
----- ( 1 ) -----
MESSA: El camp ID és obligatori.
FIELD: usuariEntitatID (ID)
TRANS: genapp.validation.required [*usuariEntitat.usuariEntitatID]
----- ( 2 ) -----
MESSA: El camp Persona és obligatori.
FIELD: usuariPersonaID (Persona)
TRANS: genapp.validation.required [*usuariEntitat.usuariPersonaID]
----- ( 3 ) -----
MESSA: El camp Entitat és obligatori.
FIELD: entitatID (Entitat)
TRANS: genapp.validation.required [*usuariEntitat.entitatID]

```

L'estructura d'una excepció de tipus WsValidationException és la següent:

Camp	Tipus	Descripció
message	String	Conté la unió de tots els missatges de validació composts i traduïts en l'idioma definit en la informació de l'usuari aplicació que realitza la cridada i separats per un retorn de carro.
faultInfo	WsValidationErrors	Classe que encapsula la informació de traducció
fieldFaults	List <WsFieldValidationError>	Conté els codis i paràmetres per traduir manualment aquest error
error	String	Conté el missatge d'error compost i traduït en l'idioma definit en la informació de l'usuari aplicació que realitza la cridada.
field	String	Conté el codi de traducció del camp que conté l'error de validació
label	String	Conté la traducció del camp que conté l'error de validació en l'idioma de l'Usuari Aplicació que ha fet la cridada.
code	String	Conté el codi de traducció del missatge
translation	Wsl18NTranslation	Conté els codis i paràmetres per traduir manualment aquest error
code	String	Conté el codi de traducció del missatge
args	Wsl18NArgument[]	Arguments del missatge
translate	boolean	Indica si aquest argument s'ha de traduir o no
value	String	Conté un valor final per cert argument el codi que s'ha de traduir.

1.7.- Constants PortaFIB

Si s'utilitza l'API descarregada de sourceforge, aquesta inclou un fitxer en el paquet `es.caib.portafib.utils.Constants` que conté totes les constants utilitzades en PortaFIB i que en ajudaran durant la programació de WebServices

Aquesta classe és una còpia de la classe Constants del mòdul utils que trobarem en el codi font de PortaFIB (`/portafib-utils/src/main/java/es/caib/portafib/utils/Constants.java`)

1.8.- Traduccions PortaFIB

Si s'utilitza l'API descarregada de sourceforge, aquesta també inclou totes les traduccions emprades en tot l'entorn de PortaFIB, la qual cosa ens permetrà gestionar les traduccions en la part del client webservice.

Podem accedir a aquestes traduccions través d'una classe anomenada `LogicI18NBaseUtils` (és una còpia de la classe `/portafib-utils/src/main/java/es/caib/portafib/utils/LogicI18NBaseUtils`)

```
// Inicialitza el sistema de traduccions
try {
    Class.forName(I18NUtils.class.getName());
} catch (Throwable th) {
    System.err.println("Error inicialitzant el sistema de traduccions : "
        + th.getMessage());
    th.printStackTrace();
}

for (Locale idioma : new Locale[]{ new Locale("ca"), new Locale("es") }) {
    System.out.println("Traducció[" + idioma.getLanguage() + "]: "
        + I18NUtils.tradueix(idioma, "genapp.listtitle", "XXXX"));
}
```

I la sortida de l'execució del programa anterior seria una cosa com:

```
Traducció[ca]: Llistat de XXXX
Traducció[es]: Listado de XXXX
```

2.- API d'Usuaris-Aplicació

2.1.- Introducció

Aquesta API serveix per gestionar els usuaris aplicació. La llista de mètodes disponibles

són (no es mostren els mètodes comuns):

- createUsuariAplicacio()
- getUsuariAplicacio()
- addRolUserToUsuariAplicacio()
- removeRolUserToUsuariAplicacio()
- addRolAdminToUsuariAplicacio()
- removeRolAdminToUsuariAplicacio()
- deleteUsuariAplicacio()

Per tots aquests mètodes, l'usuari aplicació requereix tenir el role PFI_ADMIN.

2.2.- Mètodes de l'API d'Usuari-Aplicació

2.2.1.- createUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'acció de crearUsuariAplicacio() el que fa és donar d'alta aquest usuari en el PortaFIB.

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public void createUsuariAplicacio(
    es.caib.portafib.ws.api.v1UsuariAplicacioBean usuariAplicacioBean)
    throws WsValidationException, WsI18NException;
```

Descripció:

Aquest mètode serveix per crear un nou usuari aplicació.

Paràmetres:

- **UsuariAplicacioBean usuariAplicacioBean:** Aquesta classe serveix per descriure les dades del nou usuari aplicació. Es pot veure l'estructura en detall en el punt "7.1.-Classe UsuariAplicacioBean"

2.2.2.- getUsuariAplicacio()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public UsuariAplicacioBean getUsuariAplicacio(String usuariAplicacioID)
    throws WsI18NException;
```

Descripció:

Aquest mètode serveix per obtenir informació d'usuari aplicació a partir de l'identificador o username. El contingut de la contrasenya es substitueix per un string buit.

Paràmetres:

- **String usuariAplicacioID:** Es tracta d'un String que conté l'identificador o username de l'usuari-aplicació del qual volem informació.

Resposta:

- **UsuariAplicacioBean:** Retorna un objecte de tipus UsuariAplicacioBean amb usuariAplicacioID igual al del paràmetre. Veure punt "7.1.-Classe UsuariAplicacioBean". Retorna null si no existeix usuari-aplicació amb l'ID requerit.

2.2.3.- addRolUserToUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'execució d'aquest mètode no fa res.

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public boolean addRolUserToUsuariAplicacio(String usuariAplicacioID)
    throws WsI18NException;
```

Descripció:

Aquest mètode serveix per aplicar el role PFI_USER a un usuari-aplicació.

Paràmetres:

- **String usuariAplicacioID:** Es tracta d'un String que conté l'identificador o username de l'usuari-aplicació del qual volem assignar el role.

Resposta:

- **boolean:** Indica si PortaFIB ha pogut actualitzar el sistema intern de roles del servidor d'aplicacions per fer efectiu el canvi de rol.

2.2.4.- removeRolUserToUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'execució d'aquest mètode no fa res.

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })  
public boolean removeRolUserToUsuariAplicacio(String usuariAplicacioID)  
    throws WsI18NException;
```

Descripció:

Aquest mètode serveix per llevar el role PFI_USER a un usuari-aplicació.

Paràmetres:

- **String usuariAplicacioID:** Es tracta d'un String que conté l'identificador o username de l'usuari-aplicació del qual volem llevar el role.

Resposta:

- **Tipus:** Indica si PortaFIB ha pogut actualitzar el sistema intern de roles del servidor d'aplicacions per fer efectiu el canvi de rol.

2.2.5.- addRolAdminToUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'execució d'aquest mètode no fa res.

Prototipus:
<pre>@RolesAllowed(PFI_ADMIN) public boolean addRolAdminToUsuariAplicacio(String usuariAplicacioID) throws WsI18NException;</pre>
Descripció:
Aquest mètode serveix per aplicar el role PFI_ADMIN a un usuari-aplicació.
Paràmetres:
<ul style="list-style-type: none"> String usuariAplicacioID: Es tracta d'un String que conté l'identificador o username de l'usuari-aplicació del qual volem assignar el role.
Resposta:
<ul style="list-style-type: none"> boolean: Indica si PortaFIB ha pogut actualitzar el sistema intern de roles del servidor d'aplicacions per fer efectiu el canvi de rol.

2.2.6.- removeRolAdminToUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'execució d'aquest mètode no fa res.

Prototipus:
<pre>@RolesAllowed(PFI_ADMIN) public boolean removeRolAdminToUsuariAplicacio(String usuariAplicacioID) throws WsI18NException;</pre>
Descripció:
Aquest mètode serveix per llevar el role PFI_ADMIN a un usuari-aplicació.
Paràmetres:
<ul style="list-style-type: none"> String usuariAplicacioID: Es tracta d'un String que conté l'identificador o username de l'usuari-aplicació del qual volem llevar el role.
Resposta:
<ul style="list-style-type: none"> boolean: Indica si PortaFIB ha pogut actualitzar el sistema intern de roles del servidor d'aplicacions per fer efectiu el canvi de rol.

2.2.7.- deleteUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'execució d'aquest mètode únic que fa es desvincular u usuari-aplicació de PortaFIB.

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public void deleteUsuariAplicacio(String usuariAplicacioID)
    throws WsI18NException;
```

Descripció:

Aquest mètode serveix per eliminar un usuari-aplicació.

Paràmetres:

- **String usuariAplicacioID:** Es tracta d'un String que conté l'identificador o username de l'usuari-aplicació de l'usuari-aplicació que volem borrar.

2.2.8.- listUsuariAplicacio()



En l'entorn de la CAIB (Govern Balear) indicar que com que els usuaris aplicació es gestionen a través del sistema seycon, llavors l'execució d'aquest mètode únic que fa es desvincular u usuari-aplicació de PortaFIB.

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public List<UsuariAplicacioBean> listUsuariAplicacio(
    UsuariAplicacioFilterWs usuariAplicacioFilterWs
) throws WsI18NException;
```

Descripció:

Aquest mètode serveix per llistar tots els usuari-aplicació o llistar els usuaris aplicació segons un filtre.

Paràmetres:

- **UsuariAplicacioFilterWs usuariAplicacioFilterWs:** Es tracta d'una classe que aplica diversos filtres a la llista de Usuari-aplicació. Per veure més detalls d'aquesta classe veure el punt "7.2.-Classe UsuariAplicacioFilterWs". aquest camp sempre ha de ser no NULL. Si es desitja llistar

Resposta:

- **List<UsuariAplicacioBean>:** Retorna una llista d'objectes de tipus UsuariAplicacioBean (Veure "7.1.-Classe UsuariAplicacioBean").

3.- API Usuaris-Persona, Usuaris-Entitat i Càrrecs

Aquesta API gestiona tant usuaris persona com usuaris entitat i a més Càrrecs. Si no es té molt clar cadascun d'aquests conceptes és molt recomanable revisar el "Manual d'Usuari de PortaFIB" o s'expliquen en detall aquests tres conceptes. El llistat de mètodes que conté aquesta API són:

1. COMUNS
 - a) getEntitatID()
 - b) getVersion()
 - c) getVersionWs()
 - d) downloadFileUsingEncryptedFileID()
2. USUARI-PERSONA
 - a) getInfoFromPluginUserInfoByAdministrationID()
 - b) getInfoFromPluginUserInfoByUsername()
 - c) createUsuariPersona()
 - d) getUsuariPersona()
 - e) getUsuariPersonaIDByAdministrationID()
 - f) deleteUsuariPersona()
3. USUARI-ENTITAT
 - a) createUsuariEntitatSimple()
 - b) createUsuariEntitat()
 - c) getUsuariEntitat()
 - d) getUsuariEntitatIDInMyEntitatByAdministrationID()
 - e) getUsuariEntitatIDByAdministrationID()
 - f) activateUsuariEntitat()
 - g) deactivateUsuariEntitat()
 - h) addRoleAdministradorDeEntitat()
 - i) removeRoleAdministradorDeEntitat()
 - j) addRoleSolicitant()
 - k) removeRoleSolicitant()
 - l) deleteUsuariEntitat()
4. CARREC
 - a) createCarrecSimple()
 - b) createCarrec()
 - c) getCarrec()
 - d) getCarrecsByEntitatID()
 - e) updateAdministrationIDOfCarrec()
 - f) activateCarrec()
 - g) deactivateCarrec()
 - h) deleteCarrec()

NOTA 1: En les següents descripcions de mètodes, en el prototipus es descriurà la llista de permisos necessaris de l'usuari aplicació per poder cridar al mètode en qüestió.

NOTA 2: Els mètodes que llancin algun tipus d'excepció, significa que en certes condicions aquestes excepcions poden ser llançades, per la qual cosa el programador és l'encarregat de gestionar-les com toqui.

3.1.- Mètodes Usuari Persona

3.1.1.- getInfoFromPluginUserInfoByAdministrationID()

Prototipus:
<pre>@RolesAllowed({PFI_ADMIN}) public UsuariPersonaBean getInfoFromPluginUserInfoByAdministrationID(java.lang.String administrationID) throws WsI18NException;</pre>
Descripció:
Pel usuaris-persona que accedeixen a PortaFIB requerim donar-los d'alta, però a més necessiten un conjunt d'informació addicional con nom d'usuari (username), nif, email, nom, llinatges, ...
Paràmetres:
<ul style="list-style-type: none"> String administrationID: Es tracta del NIF de la persona que volem informació
Resposta:
<ul style="list-style-type: none"> UsuariPersonaBean: Retorna un objecte de tipus UsuariPersonaBean associat al NIF (administrationID) passat per paràmetres que conté tota la informació disponible d'aquesta persona (Veure detalls d'aquesta classe al punt "7.3.-Classe UsuariPersonaBean"). Els camps d'aquesta classe estan descrites en aquest punt "...". Aquest objecte es pot emprar per donar d'alta una Persona cridant al mètode createUsuariPersona (Veure punt "3.1.3.-createUsuariPersona()"). Llança un error si la persona associada al nif no és troba.

3.1.2.- getInfoFromPluginUserInfoByUsername()

Prototipus:
<pre>@RolesAllowed({PFI_ADMIN}) public UsuariPersonaBean getInfoFromPluginUserInfoByUsername(java.lang.String username) throws WsI18NException;</pre>
Descripció:
Pel usuaris-persona que accedeixen a PortaFIB requerim donar-los d'alta, però a més necessiten un conjunt d'informació addicional con nom d'usuari (username), nif, email, nom, llinatges, ...
Paràmetres:
<ul style="list-style-type: none"> String username: Es tracta del username de la persona que volem informació.
Resposta:
<ul style="list-style-type: none"> Veure punt 3.1.1.-getInfoFromPluginUserInfoByAdministrationID()

3.1.3.- createUsuariPersona()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public void createUsuariPersona(
    UsuariPersonaBean usuariPersonaBean)
    throws WsValidationException, WsI18NException;
```

Descripció:

Aquest mètode dona d'alta un usuari persona dins el PortaFIB. El procés normal per donar d'alta una persona és el següent:

- Cridar al mètode `getInfoFromPluginUserInfoByAdministrationID()` per obtenir informació d'una persona a partir del seu nif
- Revisar que els camps obligatoris estan plens (de vegades el plugin de informació no té dades pel camp email, però PortaFIB la requereix).
- Cridar a `createUsuariPersona()`

Paràmetres:

- **UsuariPersonaBean usuariPersonaBean:** Es d'un objecte de tipus `UsuariPersonaBean` (Veure punt "7.3.-Classe UsuariPersonaBean") utilitzat per donar d'alta una Persona.

3.1.4.- getUsuariPersona()**Prototipus:**

```
@RolesAllowed({ PFI_ADMIN })
public UsuariPersonaBean getUsuariPersona(String usuariPersonaID)
    throws WsValidationException, WsI18NException;
```

Descripció:

Aquest mètode retorna informació d'una Persona donada prèviament d'alta a PortaFIB emprant el seu username

Paràmetres:

- **String usuariPersonalID:** identificador o username de la persona.

Resposta:

- **UsuariPersonaBean:** Es retornarà un objecte de tipus `UsuariPersonaBean` (Veure punt "7.3.-Classe UsuariPersonaBean") on el camp `usuariPersonalID` de la classe sigui idèntic al paràmetre `usuariPersonalID` d'aquest mètode. Si no existeix l'usuari persona llavors es retornarà un null.

3.1.5.- getUsuariPersonalIDByAdministrationID()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN, PFI_USER}) public String getUsuariPersonaIDByAdministrationID(String administrationID) throws WsValidationException, WsI18NException;</pre>
Descripció:
Aquest mètode retorna l'identificador o username d'una Persona a partir del NIF (o identificador administratiu)
Paràmetres:
<ul style="list-style-type: none">• String administrationID: És el NIF de la persona
Resposta:
<ul style="list-style-type: none">• String: Retorna l'identificador o username d'una Persona que té NIF igual al paràmetre.

3.1.6.- deleteUsuariPersona()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void deleteUsuariPersona(String usuariPersonaID) throws WsI18NException;</pre>
Descripció:
Aquest mètode borra un usuari persona a partir del seu identificador o username.
Paràmetres:
<ul style="list-style-type: none">• String usuariPersonaID: identificador o username de la persona.

3.2.- Mètodes Usuari Entitat

3.2.1.- createUsuariEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public UsuariEntitatBean createUsuariEntitat(UsuariEntitatBean usuariEntitatBean) throws WsValidationException, WsI18NException;</pre>
Descripció:
<p>Aquest mètode dona d'alta un usuari entitat dins el PortaFIB. Requereix prèviament de la creació d'un usuari persona i com és lògic de l'existència de l'entitat a la que volem enllaçar aquesta persona. Es recomana l'ús del mètode "3.2.2.-CreateUsuariEntitatSimple()" enlloc d'aquest ja que és molt més senzill.</p>
Paràmetres:
<ul style="list-style-type: none"> UsuariEntitatBean usuariEntitatBean: És un objecte de tipus UsuariEntitatBean (Veure punt "") utilitzat per donar d'alta un Usuari-Entitat.
Resposta:
<ul style="list-style-type: none"> UsuariEntitatBean: Tornarà el mateix objecte que s'ha passat per paràmetre però inclourà l'identificador de l'Usuari-Entitat.

3.2.2.- CreateUsuariEntitatSimple()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public UsuariEntitatBean createUsuariEntitatSimple(String administrationID, String entitatID) throws WsI18NException, WsValidationException;</pre>
Descripció:
<p>Fa el mateix que el mètode createUsuariEntitat() però només requereix dos paràmetres i la resta utilitza valors per defecte. A més, enlloc de requerir l'identificador de l'Usuari-Persona se li passa el NIF per identificar la Persona.</p>
Paràmetres:
<ul style="list-style-type: none"> String administrationID: NIF de la persona que volem donar d'alta dins l'entitat String entitatID: Identificador de l'entitat on volem donar d'alta la persona.
Resposta:
<ul style="list-style-type: none"> UsuariEntitatBean: Tornarà l'objecte UsuariEntitatBean amb la informació del recent creat Usuari-Entitat incloent l'identificador del propi Usuari-Entitat.

3.2.3.- getUsuariEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public UsuariEntitatBean getUsuariEntitat(String usuariEntitatID);</pre>
Descripció:
Retorna l'usuari entitat associat a l'identificador.
Paràmetres:
<ul style="list-style-type: none"> String usuariEntitatID: identificador de l'usuari entitat a cercar informació.
Resposta:
<ul style="list-style-type: none"> UsuariEntitatBean: Tornarà un objecte UsuariEntitatBean (Veure punt 7.4.-Classe UsuariEntitatBean) amb la informació de l'Usuari-Entitat que té per identificador el paràmetre passat a aquesta funció. Si no existeix llavors retornarà null.

3.2.4.- getUsuariEntitatIDInMyEntitatByAdministrationID()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN, PFI_USER }) public java.lang.String getUsuariEntitatIDInMyEntitatByAdministrationID(java.lang.String administrationID) throws WsI18NException;</pre>
Descripció:
Cerca l'identificador de l'Usuari-Entitat associat al NIF passat per paràmetre (administrationID) i retorna el seu identificador d'usuari-entitat dins l'entitat del pròpi usuari-aplicació.
Paràmetres:
<ul style="list-style-type: none"> String administrationID: NIF de l'usuari entitat que volem cercar
Resposta:
<ul style="list-style-type: none"> String: identificador d'usuari-entitat associat al NIF passat per paràmetre. Tornarà null si no troba cap usuari-entitat amb aquell nif.

3.2.5.- getUsuariEntitatIDByAdministrationID()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public String getUsuariEntitatIDByAdministrationID(String administrationID, String entitatID) throws WsI18NException;</pre>
Descripció:
Cerca l'identificador de l'Usuari-Entitat associat al NIF passat per paràmetre (administrationID) i que a més es troba dins l'entitat passada per paràmetre.
Paràmetres:
<ul style="list-style-type: none">• String administrationID: NIF de la persona que estan cercant• String entitatID: entitat de la que volem que sigui la persona que cerquem.
Resposta:
<ul style="list-style-type: none">• String: identificador d'usuari-entitat associat al NIF passat per paràmetre i que es troba dins l'entitat passada per paràmetre. Tornarà null si no troba cap usuari-entitat amb aquell NIF dins l'entitat.

3.2.6.- getUsuariEntitatIDInMyEntitatByUsuariPersonalID()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN, PFI_USER }) public java.lang.String getUsuariEntitatIDInMyEntitatByUsuariPersonalID(String usuariPersonalID, // username) throws WsI18NException;</pre>
Descripció:
Cerca l'identificador de l'Usuari-Entitat associat a l'username passat per paràmetre (usuariPersonalID) i retorna el seu identificador d'usuari-entitat dins l'entitat del pròpi l'usuari-aplicació.
Paràmetres:
<ul style="list-style-type: none">• String usuariPersonalID: Username o identificador de persona de l'usuari entitat que volem cercar
Resposta:
<ul style="list-style-type: none">• String: identificador d'usuari-entitat associat al username passat per paràmetre. Tornarà null si no troba cap usuari-entitat amb aquell username.

3.2.7.- getUsuariEntitatIDByUsuariPersonaID()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public String getUsuariEntitatIDByUsuariPersonaID(String usuariPersonaID, // username String entitatID) throws WsI18NException;</pre>
Descripció:
Cerca l'identificador de l'Usuari-Entitat associat al username passat per paràmetre (username o identificador de la persona) i que a més es troba dins l'entitat passada per paràmetre.
Paràmetres:
<ul style="list-style-type: none">• String usuariPersonaID: Username de la persona de la qual estam cercant el seu identificador d'usuari-entitat.• String entitatID: entitat de la que volem que sigui la persona que cerquem.
Resposta:
<ul style="list-style-type: none">• String: identificador d'usuari-entitat associat al usuari-persona o username passat per paràmetre i que es troba dins l'entitat passada per paràmetre. Tornarà null si no troba cap usuari-entitat amb aquell username dins l'entitat.

3.2.8.- activateUsuariEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void activateUsuariEntitat(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Activa un usuari entitat.
Paràmetres:
<ul style="list-style-type: none">• String usuariEntitatID: Identificador de l'usuari-entitat que volem activar

3.2.9.- deactivateUsuariEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void deactivateUsuariEntitat(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Desactiva un usuari-entitat.
Paràmetres:
<ul style="list-style-type: none"> String usuariEntitatID: Identificador de l'usuari-entitat que volem desactivar

3.2.10.- addRoleAdministradorDeEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void addRoleAdministradorDeEntitat(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Assigna el Role Administrador d'Entitat a cert usuari-entitat.
Paràmetres:
<ul style="list-style-type: none"> String usuariEntitatID: Identificador de l'usuari-entitat que volem assignar el role

3.2.11.- removeRoleAdministradorDeEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void <u>removeRoleAdministradorDeEntitat</u>(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Elimina el Role Administrador d'Entitat a cert usuari-entitat.
Paràmetres:
<ul style="list-style-type: none"> String usuariEntitatID: Identificador de l'usuari-entitat del que volem eliminar el role

3.2.12.- addRoleSolicitant()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void addRoleSolicitant(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Assigna el Role Sol·licitant a cert usuari-entitat.
Paràmetres:
<ul style="list-style-type: none">String usuariEntitatID: Identificador de l'usuari-entitat que volem assignar el role

3.2.13.- removeRoleSolicitant()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void removeRoleSolicitant(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Elimina el Role Sol·licitant a cert usuari-entitat.
Paràmetres:
<ul style="list-style-type: none">String usuariEntitatID: Identificador de l'usuari-entitat del que volem eliminar el role

3.2.14.- deleteUsuariEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void deleteUsuariEntitat(String usuariEntitatID) throws WsI18NException;</pre>
Descripció:
Elimina un usuari-entitat.
Paràmetres:
<ul style="list-style-type: none">String usuariEntitatID: Identificador de l'usuari-entitat que volem eliminar

3.3.- Mètodes Càrrec

3.3.1.- createCarrecSimple()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN })
public CarrecWs createCarrecSimple(
    String administrationID, String entitatID,
    String singleCarrecID, String carrecName
) throws WsValidationException, WsI18NException;
```

Descripció:

Crea un càrrec de forma simple.

Paràmetres:

- **String administrationID:** NIF de la persona a la que volem associar aquest càrrec
- **String entitatID:** Entitat en la que volem definir aquest càrrec
- **String carrecUsername:** Sufix de l'identificador de càrrec. Un identificador de càrrec està format per una cadena de caràcters formada per tres parts:
 - ◆ ID de l'entitat
 - ◆ El caràcter '_'
 - ◆ Identificador simple del càrrec o carrecUsername

Per exemple, podríem definir el càrrec el president de la CAIB, que l'identificador de càrrec seria alguna cosa com caib_carrecpresident, d'aquest l'entitat seria 'caib' i carrecUsername valdria "carrecpresident"¹.

- **String carrecName:** És el nom del Càrrec. Seguin l'exemple anterior, per exemple podríem anomenar el "carrec_president" com "President de la CAIB".

Resposta:

- **Tipus:** Retorna un objecte CarrecWs (Veure punt "7.5.-Classe CarrecWs") que conté l'identificador de càrrec final.

¹ S'aconsella no posar identificadors de càrrec molt senzills, és a dir, millor posar un identificador de càrrec "caib_carrec_secretari" que no "caib_secretari" ja que els identificadors d'usuari-entitat i els de càrrecs conviuen internament en el mateix lloc, per la qual cosa s'ha d'intentar evitar solapaments entre identificadors d'usuaris-entitat i de càrrecs.

3.3.2.- createCarrec()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void createCarrec(CarrecWs carrec) throws WsValidationException, WsI18NException;</pre>
Descripció:
Crea un càrrec a partir d'una definició completa de càrrec emprant la classe CarrecWs (Veure punt "7.5.-Classe CarrecWs")
Paràmetres:
<ul style="list-style-type: none">• CarrecWs carrec: objecte que descriu completament el càrrec a crear.

3.3.3.- getCarrec()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN, PFI_USER }) public CarrecWs getCarrec(String carrecID);</pre>
Descripció:
Retorna informació d'un càrrec a partir del seu identificador.
Paràmetres:
<ul style="list-style-type: none">• Parametre: identificador del càrrec
Resposta:
<ul style="list-style-type: none">• CarrecWs: Informació d'un càrrec a partir del seu identificador (Veure punt "7.5.-Classe CarrecWs")

3.3.4.- getCarrecsByEntitatID()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN}) public List<CarrecWs> getCarrecsByEntitatID(String entitatID) throws WsI18NException;</pre>
Descripció:
Retorna tots els càrrecs de certa entitat.
Paràmetres:
<ul style="list-style-type: none"> String entitatID: Identificador de l'entitat de la que volem obtenir la llista de càrrecs definits.
Resposta:
<ul style="list-style-type: none"> List<CarrecWs>: llista de càrrecs de l'entitat passada per paràmetre o si no hi ha càrrecs en aquella entitat llavors retorna un llista buida. Llançara una excepció si l'entitat no existeix .

3.3.5.- getCarrecsOfMyEntitat()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN, PFI_USER}) public List<CarrecWs> getCarrecsOfMyEntitat();</pre>
Descripció:
Retorna tots els càrrecs de l'entitat del propi usuari-aplicació.
Resposta:
<ul style="list-style-type: none"> List<CarrecWs>: llista de càrrecs de l'entitat del propi usuari-aplicació o si no hi ha càrrecs en l'entitat llavors retorna una llista buida.

3.3.6.- updateAdministrationIDOfCarrec()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN }) public void updateAdministrationIDOfCarrec(String carrecID, String administrationID) throws WsValidationException, WsI18NException,</pre>
Descripció:
Actualitza la persona que ostenta un càrrec.
Paràmetres:
<ul style="list-style-type: none"> String carrecID: Identificador del càrrec que volem actualitzar String administrationID: NIF de la nova persona que ocuparà aquest càrrec.

3.3.7.- activateCarrec()

Prototipus:
@RolesAllowed({ <i>PFI_ADMIN</i> }) public void activateCarrec(String carrecID) throws WsI18NException;
Descripció:
Activa un càrrec.
Paràmetres:
<ul style="list-style-type: none">• String carrecID: Identificador del càrrec que volem activar

3.3.8.- deactivateCarrec()

Prototipus:
@RolesAllowed({ <i>PFI_ADMIN</i> }) public void deactivateCarrec(String carrecID) throws WsI18NException;
Descripció:
Desactiva un càrrec.
Paràmetres:
<ul style="list-style-type: none">• String carrecID: Identificador del càrrec que volem desactivar

3.3.9.- deleteCarrec()

Prototipus:
@RolesAllowed({ <i>PFI_ADMIN</i> }) public void deleteCarrec(String carrecID) throws WsI18NException;
Descripció:
Borra un càrrec.
Paràmetres:
<ul style="list-style-type: none">• String carrecID: Identificador del càrrec que volem borrar

4.- API Petició de Firma

4.1.- Introducció

La següent API serveix per crear peticions de firma per a ser firmades per destinataris.

4.2.- Utilitats de l'API: classe PeticioDeFirmaUtils

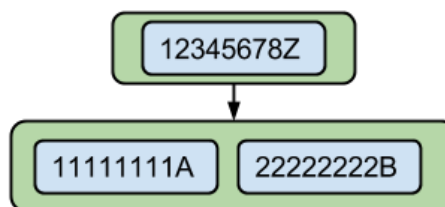
Si s'utilitza l'API generada des del projecte PortaFIB, aquesta conté una classe anomenada `es.caib.portafib.ws.api.v1.utils.PeticioDeFirmaUtils` que proporciona una sèrie de mètodes d'utilitats a l'hora de crear peticions. Aquests mètodes no creen res en el PortaFIB, només ajuden a muntar les estructures de dades a enviar al servidor.

- ***FitxerBean constructFitxerBeanFromResource(String name, String mime) throws Exception***: Construeix un element `FitxerBean` a partir d'un fitxer que es troba com a recurs (resource) d'algun jar. Representa a un fitxer PortaFIB i és el format requerit per enviar fitxers al Servidor. Més informació de `FitxerBean` en el punt "1.5.4.1.-Classe FitxerBean".
- ***FitxerBean constructFitxerBeanFromFile(File file, String mime) throws Exception***: Construeix un element `FitxerBean` a partir d'un fitxer que es troba en el sistema de fitxers. Representa a un fitxer PortaFIB i és el format requerit per enviar fitxers al Servidor. Més informació de `FitxerBean` en el punt "1.5.4.1.-Classe FitxerBean".
- ***FitxerBean constructFitxerBeanFromInputStream(String name, String mime, InputStream is) throws IOException***: Construeix un element `FitxerBean` a partir d'unes dades passades com a un flux de dades (`InputStream`). Representa a un fitxer PortaFIB i és el format requerit per enviar fitxers al Servidor. Més informació de `FitxerBean` en el punt "1.5.4.1.-Classe FitxerBean".
- ***byte[] toByteArray(InputStream input) throws IOException***: mètode d'utilitat per convertir un `InputStream` a `byte[]`.
- ***FluxDeFirmesWs constructFluxDeFirmesWs(PortaFIBUsuariEntitatWs usuariEntitatAPI, String[] nifs) throws WsI18NException***: Construeix un objecte `FluxDeFirmesWs` que representa les persones i l'estructura en que han de firmar, a partir d'una llista dels NIFs dels destinataris. Més informació de `FluxDeFirmesWs` en el punt "7.6.-Classe FluxDeFirmesWs".
- ***FluxDeFirmesWs constructFluxDeFirmesWsUsingBlocDeFirmes(PortaFIBUsuariEntitatWs usuariEntitatAPI, String[][] nifs) throws WsI18NException***: Construeix un objecte `FluxDeFirmesWs` que representa les persones i l'estructura en que han de firmar, a partir d'una llista dels NIFs dels destinataris. Els NIFs venen definits en una estructura de dues dimensions, en la primera dimensió defineix el bloc i tots els nifs de la segona dimensió es

guardaran dins el mateix bloc. Veure dibuix de del mètode `constructPeticioDeFirma()`. Més informació de `FluxDeFirmesWs` en el punt “7.6.-Classe `FluxDeFirmesWs`”.

- ***PeticioDeFirmaWs constructPeticioDeFirma(PortaFIBUsuariEntitatWs usuariEntitatAPI, String titol, String remitent, FitxerBean fitxerAFirmar, String[] nifs) throws Wsl18NException:*** Construeix un objecte de tipus `PeticioDeFirma` a partir del titol, nom del remitent (normalment el nom de l'aplicació que envia la petició), el fitxer a firmar i el conjunt de destinataris a partir d'una llista dels seus NIFs.
- ***PeticioDeFirmaWs constructPeticioDeFirma(PortaFIBUsuariEntitatWs usuariEntitatAPI, String titol, String remitent, FitxerBean fitxerAFirmar, String[][] nifs) throws Wsl18NException:*** Construeix un objecte de tipus `PeticioDeFirma` a partir del titol, nom del remitent (normalment el nom de l'aplicació que envia la petició), el fitxer a firmar i el conjunt de destinataris a partir d'una llista dels seus NIFs. Els NIFs venen definits en una estructura de dues dimensions, en la primera dimensió defineix el bloc i tots els nifs de la segona dimensió es guardaran dins el mateix bloc.

```
{
  { 12345678Z },
  { 11111111A , 22222222B }
}
```



- ***PeticioDeFirmaWs constructPeticioDeFirma(String titol, String remitent, FitxerBean fitxerAFirmar, FluxDeFirmesWs fluxDeFirmesWs):*** Construeix un objecte de tipus `PeticioDeFirma` a partir del titol, nom del remitent (normalment el nom de l'aplicació que envia la petició), el fitxer a firmar i el conjunt de destinataris definits en una estructura de tipus `FluxDeFirmesWs`. Més informació de `FluxDeFirmesWs` en el punt “7.6.-Classe `FluxDeFirmesWs`”.
- ***AnnexBean constructAnnexBeanFromResource(String name, String mime, boolean adjuntar, boolean firmar) throws Exception:*** Construeix un objecte `AnnexBean` a partir d'un fitxer guardat coma recurs (resource) dins d'un jar i de les propietats “adjuntar” i “firmar”.
- ***AnnexBean constructAnnexBeanFromFile(File name, String mime, boolean adjuntar, boolean firmar) throws Exception:*** Construeix un objecte `AnnexBean` a partir d'un fitxer guardat dins el sistema de fitxers i de les propietats “adjuntar” i “firmar”.

4.3.- Mètodes de l'API PortaFIBPeticioDeFirma

- `createPlantillaFluxDeFirmes()`
- `deletePlantillaFluxDeFirmes()`
- `instantiatePlantillaFluxDeFirmes()`
- `getCurrentCustodiaPluginClass()`

- getDefaultCustodialInfo()
- getTipusDeDocuments()
- createPeticioDeFirma()
- startPeticioDeFirma()
- createAndStartPeticioDeFirma()
- getPeticioDeFirma()
- getLastSignedFileOfPeticioDeFirma()
- getStateOfPeticioDeFirma()
- pausePeticioDeFirma()
- resetPeticioDeFirma()
- deletePeticioDeFirma()

4.3.1.- createPlantillaFluxDeFirmes()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public Long createPlantillaFluxDeFirmes(FluxDeFirmesWs fluxDeFirmesWs,
    boolean compartir) throws WsValidationException, WsI18NException;
```

Descripció:

Crea una plantilla de Flux de Firmes que opcionalment la podem compartir amb la resta de l'entitat.

Paràmetres:

- **FluxDeFirmesWs fluxDeFirmes:** Aquest objecte conté les persones que han de firmar i l'ordre amb que ho han de fer. Podeu veure el punt "7.6.-Classe FluxDeFirmesWs".
- **boolean compartir:** Indica si volem compartir aquesta Plantilla amb la resta de l'entitat. Només poden compartir els usuaris aplicació amb permís PFI_ADMIN. Pels usuaris amb només PFI_USER aquest camp serà ignora

Resposta:

- **Long:** Retorna l'identificador de la nova Plantilla.

4.3.2.- deletePlantillaFluxDeFirmes()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public void deletePlantillaFluxDeFirmes (@WebParam(name = "plantillaDeFluxDeFirmesID") long plantillaDeFluxDeFirmesID) throws WsI18NException;</pre>
Descripció:
Elimina una plantilla de Flux de Firmes
Paràmetres:
<ul style="list-style-type: none"> long plantillaDeFluxDeFirmesID: identificador de la plantilla que és vol eliminar.
Resposta:

4.3.3.- instantiatePlantillaFluxDeFirmes()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public FluxDeFirmesWs instantiatePlantillaFluxDeFirmes (long plantillaDeFluxDeFirmesID) throws WsI18NException;</pre>
Descripció:
Retorna una Plantilla de flux de firmes en forma de Flux de Firmes llest per ser utilitzar en una Petició de Firma.
Paràmetres:
<ul style="list-style-type: none"> long plantillaDeFluxDeFirmesID: identificador de la plantilla de la qual és vol obtenir una instància de flux de firmes.
Resposta:
<ul style="list-style-type: none"> FluxDeFirmesWs: instància de flux de firmes associada a la plantilla.

4.3.4.- **getCurrentCustodiaPluginClass()**

Prototipus:

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public String getCurrentCustodiaPluginClass() throws WsI18NException;
```

Descripció:

Retorna l'identificador del sistema que actualment gestiona la custòdia dels document firmats. Si val retorna null indica que no hi ha definit cap sistema de custòdia. Aquest valor serà utilitzat per assignar-ho al camp "CustodiaPluginClassID" de la classe CustodialInfoBean (Veure més detalls al punt "7.9.-Classe CustodialInfoBean")

Resposta:

- **String:** Retorna l'identificador del sistema que actualment gestiona la custòdia dels document firmats. Si val retorna null indica que no hi ha definit cap sistema de custòdia.

4.3.5.- getDefaultCustodialInfo()

<<< AQUI S'HA DE PARLAR DEL CAMP EDITABLE >>>>>

Prototipus:

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public CustodialInfoBean getDefaultCustodiaInfo(String title, String language)
throws WsI18NException;
```

Descripció:

Retorna una instància de CustodialInfoBean amb els valors per defecte.

Paràmetres:

- **String title:** Títol que identificarà aquesta informació de custòdia. Normalment s'utilitzarà el nom de la petició de firma.
- **String language:** Idioma en format "ISO 639-1". S'ha de comprovar que aquest idioma estigui suportat dins de l'entorn PortaFIB. El valors possibles són "ca" per català o "es" per castellà. S'utilitzarà pels missatges d'alguns camps de la classe CustodialInfoBean.

Resposta:

- **CustodialInfoBean:** Objecte de informació de custòdia amb els valors per defecte. Per més informació vegeu el punt "7.9.-Classe CustodialInfoBean"

4.3.6.- getTipusDeDocuments()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public List<TipusDocumentInfoWs> getTipusDeDocuments(String idioma) throws WsI18NException;</pre>
Descripció:
Retorna tots els tipus de documents disponibles en PortaFIB per l'usuari aplicació que fa la cridada al WebService
Paràmetres:
<ul style="list-style-type: none"> String idioma: (opcional) Idioma en format "ISO 639-1". S'ha de comprovar que aquest idioma estigui suportat dins de l'entorn PortaFIB. El valors possibles són "ca" per català o "es" per castellà. No pot valer null. Si aquesta cadena es buida llavors utilitzarà l'idioma de l'usuari aplicació. Serveix per traduir els noms dels tipus de documents
Resposta:
<ul style="list-style-type: none"> List<TipusDocumentInfoWs>: Llistat d'estructures TipusDocumentInfoWs que contenen l'identificador i el nom del tipus de document. Veure punt "7.10.-Classe TipusDocumentInfoWs"

4.3.7.- createPeticioDeFirma()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public PeticioDeFirmaWs createPeticioDeFirma(PeticioDeFirmaWs peticioDeFirmaWs) throws WsI18NException, WsValidationException;</pre>
Descripció:
Crea una petició de firma en el Servidor PortaFib sense posar-la en marxa.
Paràmetres:
<ul style="list-style-type: none"> PeticioDeFirmaWs peticioDeFirmaWs: Dades de la petició de firma. Veure punt "7.13.- Classe PeticioDeFirmaWs"
Resposta:
<ul style="list-style-type: none"> PeticioDeFirmaWs: retorna la mateixa petició que l'enviada però amb els identificadors assignats. Per exemple peticioDeFirmaWs.getPeticioDeFirmaID() conté l'identificador de la petició de firma que acabem de crear.

4.3.8.- startPeticioDeFirma()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public void startPeticioDeFirma(long peticioDeFirmaID)
    throws WsI18NException;
```

Descripció:

Posa en marxa una petició de firma no iniciada. També serveix per reiniciar una petició de firma pausada.

Paràmetres:

- **long peticioDeFirmaID:** identificador de la petició de firma que volen posar en marxa o volen reiniciar.

4.3.9.- createAndStartPeticioDeFirma()**Prototipus:**

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public PeticioDeFirmaWs createAndStartPeticioDeFirma(
    PeticioDeFirmaWs peticioDeFirmaWs)
    throws WsI18NException, WsValidationException,
```

Descripció:

Crea i inicia una petició de firma. Seria la conjunció dels mètodes “4.3.7.- createPeticioDeFirma()” i “4.3.8.-startPeticioDeFirma()”

Paràmetres:

- **PeticioDeFirmaWs peticioDeFirmaWs:** Dades de la petició de firma. Veure punt “7.13.- Classe PeticioDeFirmaWs”

Resposta:

- **PeticioDeFirmaWs:** retorna la mateixa petició que l'enviada però amb el identificadors assignats. Per exemple peticioDeFirmaWs.getPeticioDeFirmaID() conté l'identificador de la petició de firma que acabem de crear.

4.3.10.- getPeticioDeFirma()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public PeticioDeFirmaWs getPeticioDeFirma(long peticioDeFirmaID)
    throws WsI18NException
```

Descripció:

Obté informació d'una petició de firma a partir del seu identificador. Tornarà null si la petició no existeix.

Paràmetres:

- **long peticioDeFirmaID:** identificador de la petició de firma de la que volen obtenir informació.

Resposta:

- **PeticioDeFirmaWs:** retorna informació de la petició de firma amb identificador "peticioDeFirmaID" emprant una estructura PeticioDeFirmaWs (Veure detall al punt "7.13.- Classe PeticioDeFirmaWs"). Tornarà null si la petició no existeix.

4.3.11.- getLastSignedFileOfPeticioDeFirma()

Prototipus:

```
@RolesAllowed({ PFI_ADMIN , PFI_USER })
public FitxerBean getLastSignedFileOfPeticioDeFirma(long peticioDeFirmaID)
    throws WsI18NException
```

Descripció:

Retorna un cosa diferent segons els casos següents:

- El darrer fitxer firmat si la petició de firma esta en marxa i algú ha firmat
- Els fitxer adaptat si la petició de firma esta en marxa i ningú ha firmat (només si el tipus de firma és PADES)
- El fitxer original (fitxer a firmar) si la petició de firma no s'ha iniciat (o si s'ha iniciat però el tipus de firma és diferent de PADES)

Paràmetres:

- **long peticioDeFirmaID:** identificador de la petició de firma de la que volen obtenir el darrer fitxer firmat.

Resposta:

- **FitxerBean:** Estructura FitxerBean que encapsula un fitxer. Veure Descripció per més detalls.

4.3.12.- getStateOfPeticioDeFirma()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public Integer getStateOfPeticioDeFirma(long peticioDeFirmaID) throws WsI18NException</pre>
Descripció:
Retorna l'estat en que és troba la petició de firma. Tornarà null si la petició no existeix.
Paràmetres:
<ul style="list-style-type: none"> • long peticioDeFirmaID: identificador de la petició de firma de la que volen obtenir el seu estat.
Resposta:
<ul style="list-style-type: none"> • Tipus: Un sencer que identifica l'estat en que es troba la petició de firma. El significat de cada valor es descriu en la següent llista: <ul style="list-style-type: none"> ◆ NULL: la petició no existeix. ◆ Constants.TIPUSESTATPETICIODEFIRMA_NOINICIAT = 0: No iniciada. ◆ Constants.TIPUSESTATPETICIODEFIRMA_ENPROCES = 1: En procés. ◆ Constants.TIPUSESTATPETICIODEFIRMA_PAUSAT = 2: Pausada ◆ Constants.TIPUSESTATPETICIODEFIRMA_REBUTJAT = 3: Rebutjada ◆ Constants.TIPUSESTATPETICIODEFIRMA_FIRMAT = 4: Firmada

4.3.13.- pausePeticioDeFirma()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public void pausePeticioDeFirma(long peticioDeFirmaID) throws WsI18NException</pre>
Descripció:
Pausa una petició de Firma. Comentar que les firmes dels bloc de firmes actius no es pausen, però si que una petició pausada no processa noves firmes dels blocs de firmes següents.
Paràmetres:
<ul style="list-style-type: none"> • long peticioDeFirmaID: identificador de la petició de firma de la que volen obtenir el seu estat.

4.3.14.- resetPeticioDeFirma()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public PeticioDeFirmaWs resetPeticioDeFirma(long peticioDeFirmaID) throws WsI18NException</pre>
Descripció:
Reinicialitza una petició de firma a l'estat No Iniciat.
Paràmetres:
<ul style="list-style-type: none"> long peticioDeFirmaID: identificador de la petició de firma que volem reinicialitzar.
Resposta:
<ul style="list-style-type: none"> PeticioDeFirmaWs: La mateixa petició de firma amb els nous identificadors. Null si la petició no existeix.

4.3.15.- deletePeticioDeFirma()

Prototipus:
<pre>@RolesAllowed({ PFI_ADMIN , PFI_USER }) public void deletePeticioDeFirma(long peticioDeFirmaID) throws WsI18NException;</pre>
Descripció:
Elimina una petició de Firma.
Paràmetres:
<ul style="list-style-type: none"> long peticioDeFirmaID: identificador de la petició de firma que volem borrar.

4.4.- Exemples

El següent exemple mostra la creació d'una petició amb annexe i custòdia. L'enviament al servidor i la posada en marxa, per després descarregar el fitxer adaptat i finalment borrar la petició.

```
package es.caib.portafib.ws.v1.test;

import java.io.File;
import java.util.Map;

import javax.xml.ws.BindingProvider;

import es.caib.portafib.utils.Constants;
import es.caib.portafib.ws.api.v1.AnnexBean;
import es.caib.portafib.ws.api.v1.CustodiaInfoBean;
import es.caib.portafib.ws.api.v1.FitxerBean;
import es.caib.portafib.ws.api.v1.PeticioDeFirmaWs;
import es.caib.portafib.ws.api.v1.PortaFIBPeticioDeFirmaWs;
import es.caib.portafib.ws.api.v1.PortaFIBPeticioDeFirmaWsService;
```

```
import es.caib.portafib.ws.api.v1.PortaFIBUsuariEntitatWs;
import es.caib.portafib.ws.api.v1.PortaFIBUsuariEntitatWsService;
import es.caib.portafib.ws.api.v1.WsI18NException;
import es.caib.portafib.ws.api.v1.WsValidationException;
import es.caib.portafib.ws.api.v1.utils.PeticioDeFirmaUtils;
import es.caib.portafib.ws.api.v1.utils.WsClientUtils;

public class Test {

    public static void main(String[] args) {

        try {

            String usr_app = "fundaciobit_usrapp"; // username application
            String pwd_app = "fundaciobit_usrapp"; // password application
            // Adreça servidor
            final String endpointBase = "http://localhost:8080/portafib/ws/v1/";

            PortaFIBPeticioDeFirmaWs api;
            {
                String endpoint = endpointBase + "PortaFIBPeticioDeFirma";
                PortaFIBPeticioDeFirmaWsService service;
                URL wsdl = new URL(endpoint + "?wsdl");
                service = new PortaFIBPeticioDeFirmaWsService(wsdl);
                api = service.getPortaFIBPeticioDeFirmaWs();

                Map<String, Object> reqContext;
                reqContext = ((BindingProvider) api).getRequestContext();
                reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, endpoint);
                reqContext.put(BindingProvider.USERNAME_PROPERTY, usr_app);
                reqContext.put(BindingProvider.PASSWORD_PROPERTY, pwd_app);
            }

            PortaFIBUsuariEntitatWs apiUE;
            {
                String endpoint = endpointBase + "PortaFIBUsuariEntitat";
                PortaFIBUsuariEntitatWsService service;
                URL wsdl = new URL(endpoint + "?wsdl");
                service = new PortaFIBUsuariEntitatWsService(wsdl);
                apiUE = service.getPortaFIBUsuariEntitatWs();

                Map<String, Object> reqContext;
                reqContext = ((BindingProvider) apiUE).getRequestContext();
                reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, endpoint);
                reqContext.put(BindingProvider.USERNAME_PROPERTY, usr_app);
                reqContext.put(BindingProvider.PASSWORD_PROPERTY, pwd_app);
            }

            System.out.println("Versio " + api.getVersion());
            System.out.println("Idiomes:" +
                Arrays.toString(api.getSupportedLanguages().toArray()));

            final String titol = "Peticio de Test amb Custodia";
            final String remitent = "Helium";

            // Existeix sistema de custodia
            CustodiaInfoBean custodiaInfoBean = null;
        }
    }
}
```

```
String pluginClassID = api.getCurrentCustodiaPluginClass();
if (pluginClassID == null) {
    System.out.println("Avis: No hi ha sistema de custodia definit"
        + " o l'usuari aplicació " + usr_app + " no pot custodiar");
} else {
    // Obtener sistema de custodia
    // Si language és un string buit s'assigna l'idioma del usuariApp
    final String lang = "";
    custodiaInfoBean = api.getDefaultCustodiaInfo(titol, lang);
}

// Annexes
AnnexBean annex = PeticioDeFirmaUtils.constructAnnexBeanFromFile(
    new File("c:\\annex.txt"), "text/plain", true, true);

// Fitxer a Firmar
FitxerBean fitxerAFirmar;
fitxerAFirmar = PeticioDeFirmaUtils.constructFitxerBeanFromFile(
    new File("c:\\test.pdf"), Constants.PDF_MIME_TYPE);

String[] nifs = new String[] { "11223344T" };

// Crear Peticio
PeticioDeFirmaWs peticioDeFirmaWs;
peticioDeFirmaWs = PeticioDeFirmaUtils.constructPeticioDeFirma(
    apiUE, titol, remitent, fitxerAFirmar, nifs);

// Assignar Annexes a la Peticio
peticioDeFirmaWs.getAnnexs().add(annex);

// Assignar custodia a la Peticio
peticioDeFirmaWs.setCustodiaInfo(custodiaInfoBean);

Long peticioDeFirmaID = null;
try {
    // Crear peticio
    peticioDeFirmaWs = api.createPeticioDeFirma(peticioDeFirmaWs);
    peticioDeFirmaID = peticioDeFirmaWs.getPeticioDeFirmaID();
    System.out.println("Creada peticio amb ID = " + peticioDeFirmaID);

    // Arrancar
    api.startPeticioDeFirma(peticioDeFirmaID);

    // Obtener petició de Firma
    peticioDeFirmaWs = api.getPeticioDeFirma(peticioDeFirmaID);

    // Imprimir estat
    System.out.println("Estat de la peticio = "
        + peticioDeFirmaWs.getTipusEstatPeticioDeFirmaID());

    // Decaregar Document adaptat o esperar a que
    // la petició finalitzi pel doc firmat

    // Info document firmat
    FitxerBean info;
    info = api.getLastSignedFileOfPeticioDeFirma(peticioDeFirmaID);
```



```

// Descarregar document
FitxerBean fdoc;
fdoc = api.downloadFileUsingEncryptedFileID(info.getEncryptedFileID());

byte[] data = fdoc.getData();

} finally {
    if (peticioDeFirmaID != null) {
        // Borrar la petició
        api.deletePeticioDeFirma(peticioDeFirmaID);
    }
}

} catch (WsI18NException i18ne) {
    System.err.println(WsClientUtils.toString(i18ne));
} catch (WsValidationException ve) {
    System.err.println(WsClientUtils.toString(ve));
} catch (Exception e) {
    System.err.println("Error desconegut: " + e.getMessage());
}

}
}




```

5.- API Callback WS PortaFIB v1

5.1.- Introducció

L'API de Callback serveix al servidor PortaFIB per enviar notifikacions via webservices dels events que ocorren en peticions relacionades amb els usuaris aplicació, és a dir, les peticions de firma enviades via WS emprant un usuari-aplicació. El funcionament normal és que des del lloc (aplicació externa) on es fa la petició de firma via webservices sigui el lloc on es posi un servidor webservices escoltant amb aquesta API.

Com és lògic en el servidor PortaFIB s'haurà definit en l'usuari aplicació el camp "URL de Callback" que apuntarà a aquest servidor de webservices.

fundaciobit_usrapp	PortaFIB Callback v1.0	http://localhost:8080/portafib/cb/v1 /PortaFIBCallback	✓	PFI_USER	  
--------------------	---------------------------	---	---	----------	---

A continuació és mostra una captura de l'entorn web associat al Callback d'exemple, que l'únic que fa es guardar els events rebuts i mostrar-los via web. En el meu cas tenc el servidor en localhost:8080, llavors la URL a aquest entorn d'exemple és la següent: "http://localhost:8080/portafib/portafibcb/":

Llistat de Callbacks Rebuts (WSDL) Fer Net

Data	Ver.	Entitat	Applicació	Tipus	Petició	Firma	Actor
26/01/2016 08:35:38	1	fundaciobit	fundaciobit_usrapp	REBUTJAT	ID:174934 Titol: Peticio de Test amb Custodia - 93715 Estat: REBUTJAT Info: null Rebuig: No és correcte	-	ID: fundaciobit_anadal NIF: 68456845C Name: Antoni Nadal Bennasar
26/01/2016 08:35:22	1	fundaciobit	fundaciobit_usrapp	ENPROCES	ID:174934 Titol: Peticio de Test amb Custodia - 93715 Estat: EN_PROCES Info: null	-	-
26/01/2016 08:29:03	1	fundaciobit	fundaciobit_usrapp	FIRMADA	ID:174865 Titol: Peticio de Test amb Custodia - 91323 Estat: FIRMAT Info: null Custodia: http://localhost:8080/custodia/index.jsp?hash=a3c007f144f0bdb6ae3d085a2f40cd9	-	-
26/01/2016 08:29:03	1	fundaciobit	fundaciobit_usrapp	FIRMA PARCIAL	ID:174865 Titol: Peticio de Test amb Custodia - 91323 Estat: EN_PROCES Info: null Rebuig:	ID: 174863 S/N: 1023931205 Issuer: C=ES,O=FNMT,OU=FNMT Clase 2 CA Subject: C=es,O=FNMT,OU=FNMT Clase 2 CA,OU=500070015,CN=NOMBRE NADAL ANTONI - NIF 68456845C	ID: fundaciobit_anadal NIF: 68456845C Name: Antoni Nadal
26/01/2016 08:28:35	1	fundaciobit	fundaciobit_usrapp	ENPROCES	ID:174865 Titol: Peticio de Test amb Custodia - 91323 Estat: EN_PROCES Info: null	-	-
26/01/2016 08:28:08	1	fundaciobit	fundaciobit_usrapp	PAUSAT	ID:174865 Titol: Peticio de Test amb Custodia - 91323 Estat: PAUSAT Info: null	-	-
26/01/2016 08:27:36	1	fundaciobit	fundaciobit_usrapp	ENPROCES	ID:174865 Titol: Peticio de Test amb Custodia - 91323 Estat: EN_PROCES Info: null	-	-

5.2.- Crear WebService

El servidor webservices es pot crear a partir del projecte ws\portafib_callback_server del codi font de PortaFIB que genera un jar amb webservices per un servidor JBoss amb CXF. Només s'hauran de modificar alguns noms de projecte dins el pom i dels paquets java, així com la implementació dels mètodes.

Podeu crear també el servei webservice a partir del wsdl que trobareu en la següent ruta del codi font de PortaFIB ws\portafib_callback_apí\PortaFIBCallBack_v1.wsdl. Adjuntam una còpia per si no es té accés a aquest arxiu.

```
<?xml version='1.0' encoding='UTF-8'?><wsdl:definitions name="PortaFIBCallBackWsService"
targetNamespace="http://v1.server.callback.ws.portafib.caib.es/"
xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://v1.server.callback.ws.portafib.caib.es/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xs:schema attributeFormDefault="unqualified" elementFormDefault="unqualified"
targetNamespace="http://v1.server.callback.ws.portafib.caib.es/"
xmlns:tns="http://v1.server.callback.ws.portafib.caib.es/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:element name="event" type="tns:event" />
      <xs:element name="eventResponse" type="tns:eventResponse" />
      <xs:element name="getVersionWs" type="tns:getVersionWs" />
      <xs:element name="getVersionWsResponse" type="tns:getVersionWsResponse" />
    </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

<xs:complexType name="getVersionWs">
<xs:sequence />
</xs:complexType>
<xs:complexType name="getVersionWsResponse">
<xs:sequence>
<xs:element name="return" type="xs:int" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="event">
<xs:sequence>
<xs:element minOccurs="0" name="event" type="tns:portaFIBEvent" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="portaFIBEvent">
<xs:sequence>
<xs:element minOccurs="0" name="actor" type="tns:actor" />
<xs:element minOccurs="0" name="applicationID" type="xs:string" />
<xs:element minOccurs="0" name="entityID" type="xs:string" />
<xs:element minOccurs="0" name="eventDate" type="xs:dateTime" />
<xs:element name="eventTypeID" type="xs:int" />
<xs:element minOccurs="0" name="sign" type="tns:sign" />
<xs:element minOccurs="0" name="signingRequest" type="tns:signingRequest" />
<xs:element name="version" type="xs:int" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="actor">
<xs:sequence>
<xs:element minOccurs="0" name="administrationID" type="xs:string" />
<xs:element minOccurs="0" name="ID" type="xs:string" />
<xs:element minOccurs="0" name="name" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="sign">
<xs:sequence>
<xs:element name="ID" type="xs:long" />
<xs:element minOccurs="0" name="issuer" type="xs:string" />
<xs:element minOccurs="0" name="serialNumber" type="xs:integer" />
<xs:element minOccurs="0" name="subject" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="signingRequest">
<xs:sequence>
<xs:element minOccurs="0" name="additionalInformation" type="xs:string" />
<xs:element minOccurs="0" name="custodyURL" type="xs:string" />
<xs:element name="ID" type="xs:long" />
<xs:element minOccurs="0" name="rejectionReason" type="xs:string" />
<xs:element name="state" type="xs:int" />
<xs:element minOccurs="0" name="title" type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventResponse">
<xs:sequence />
</xs:complexType>
<xs:element name="CallBackFault" type="tns:CallBackFault" />
<xs:complexType name="CallBackFault">
<xs:sequence />
</xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="getVersionWs">
  <wsdl:part element="tns:getVersionWs" name="parameters">
    </wsdl:part>
  </wsdl:part>

```

```

</wsdl:message>
<wsdl:message name="CallbackException">
  <wsdl:part element="tns:CallbackFault" name="CallbackException">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="eventResponse">
  <wsdl:part element="tns:eventResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="event">
  <wsdl:part element="tns:event" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getVersionWsResponse">
  <wsdl:part element="tns:getVersionWsResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="PortaFIBCallbackWs">
  <wsdl:operation name="getVersionWs">
    <wsdl:input message="tns:getVersionWs" name="getVersionWs">
    </wsdl:input>
    <wsdl:output message="tns:getVersionWsResponse" name="getVersionWsResponse">
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="event">
    <wsdl:input message="tns:event" name="event">
    </wsdl:input>
    <wsdl:output message="tns:eventResponse" name="eventResponse">
    </wsdl:output>
    <wsdl:fault message="tns:CallbackException" name="CallbackException">
    </wsdl:fault>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PortaFIBCallbackWsServiceSoapBinding" type="tns:PortaFIBCallbackWs">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="getVersionWs">
    <soap:operation soapAction="" style="document" />
    <wsdl:input name="getVersionWs">
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="getVersionWsResponse">
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="event">
    <soap:operation soapAction="" style="document" />
    <wsdl:input name="event">
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output name="eventResponse">
      <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="CallbackException">
      <soap:fault name="CallbackException" use="literal" />
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="PortaFIBCallbackWsService">
  <wsdl:port binding="tns:PortaFIBCallbackWsServiceSoapBinding" name="PortaFIBCallbackWs">
    <soap:address location="http://localhost:8080/portafib/cb/v1/PortaFIBCallback" />
  </wsdl:port>
</wsdl:service>

```

```
</wsdl:definitions>
```

5.3.- API de Callback

Aquesta API de Callback només consta de dos mètodes:

```
@WebService
public interface PortaFIBCallbackWs {

    @WebMethod
    public int getVersionWs();

    @WebMethod
    public void event(@WebParam(name = "event") PortaFIBEvent event) throws CallbackException;

}
```

El primer, *getVersionWs()*, és només un mètode per versionar els WS. Ha de retornar 1.

El segon, *event(...)*, és el que utilitza PortaFIB per notificar als clients canvis en el procés de firma que els Usuaris-Aplicació han enviat a PortaFIB:

Prototipus:

```
public void event(@WebParam(name = "event") PortaFIBEvent event) throws
CallbackException;
```

Descripció:

Rep les notificacions de canvis en algun procés de firma que ha enviat algun Usuari-Aplicació al que se li ha associat aquest WS per rebre informació.

Paràmetres:

- **PortaFIBEvent event:** Bean que conté tota la informació d'algun event o canvi en el procés de firma. Veure punt "5.4.1.-Classe PortalFIBEvent" per més detalls.

Resposta:

- No respon res al servidor. Això indica que tot ha anat correctament. Si és vol informar a PortaFIB que aquest Callback no accepta la notificació o no ha pogut processar-la, llavors s'ha de realitzar llançant una excepció de tipus *CallbackException* (Veure següent apartat)

Excepcions:

- Per informar a PortaFIB que aquest Callback no accepta la petició o durant el procés d'aquesta notificació s'ha produït un error, llavors s'ha de realitzar llançant una excepció de tipus *CallbackException*. Exemple d'ús:

```
throw new CallbackException("Error processant notificació");
```

5.4.- Classes WS Callback

5.4.1.- Classe PortalFIBEvent

Els camps de l'estructura 5.2.1.- PortalFIBEvent són els següents:

Nom	Tipus	Descripció
eventTypeID	int	Indica el tipus d'event. El llistat de tipus es descriu en el punt "5.4.1.1.-Taula de tipus d'events"
version	int	Versió del WS del Servidor
eventDate	Date	Data en que s'ha produït l'event
applicationID	String	És l'aplicació a qui va dirigida la informació, és a dir, la que ha realitzat la Petició de Firma
entityID	String	És l'entitat o organització
signingRequest	SigningRequest	Informació sobre la Petició De Firma. Els detalls d'aquesta classe es mostren al punt "5.4.2.-Classe SigningRequest"
sign	Sign	Informació sobre la firma. Aquest camp té valor només quan el tipus d'event és NOTIFICACIOAVIS_FIRMA_PARCIAL. Podeu trobar més detalls al punt "5.4.3.-Classe Sign"
actor	Actor	És la persona que ha realitzat l'acció. Aquest camp té valor només quan el tipus d'event és NOTIFICACIOAVIS_FIRMA_PARCIAL o NOTIFICACIOAVIS_PETICIO_REBUTJADA, que indica qui ha realitzat la firma o qui ha rebutjat la petició. Més informació al següent punt "5.4.4.-Classe Actor"

5.4.1.1.- Taula de tipus d'events

A continuació es descriuen els tipus d'events que existeixen. Es important comentar que aquests events només es llancen per part del servidor de PortaFIB sobre les Peticions de Firma que han sigut enviades a traves de WS enprant Usuaris-Aplicació.

Valor	Nom/Descripció
0	NOTIFICACIOAVIS_PETICIO_EN_PROCES: Indica que una Petició de Firma s'ha posat s'ha iniciat o reiniciat després de ser pausada.
50	NOTIFICACIOAVIS_FIRMA_PARCIAL: Indica si un usuari ha realitzat una firma. Com que en una petició de Firma hi pot haver múltiples destinataris (i els delegats associats), aquesta notificació només informa que un d'aquest destinataris ha signat.
60	NOTIFICACIOAVIS_PETICIO_FIRMADA: Indica que la petició de firma ha finalitzat correctament, és a dir, s'han realitzat totes les firmes requerides.
70	NOTIFICACIOAVIS_PETICIO_REBUTJADA: Indica que una petició ha sigut rebutjada per algun dels destinataris o delegats.

80	NOTIFICACIOAVIS_PETICIO_PAUSADA: Indica que una petició ha sigut pausada.
----	---

5.4.2.- Classe SigningRequest

Els camps d'aquesta classe són els següents:

Nom	Tipus	Descripció
iD	long	Identificador de la Petició de Firma
title	String	Títol de la petició de firma
state	int	Estat de la petició de firma. Els valors poden ser: <ul style="list-style-type: none"> TIPUSESTATPETICIODEFIRMA_NOINICIAT = 0; TIPUSESTATPETICIODEFIRMA_ENPROCES = 1; TIPUSESTATPETICIODEFIRMA_PAUSAT = 2; TIPUSESTATPETICIODEFIRMA_REBUTJAT = 3; TIPUSESTATPETICIODEFIRMA_FIRMAT = 4;
rejectionReason	String	Quan la petició es rebutjada aquest camp conté la raó de rebuig.
additionalInformation	String	Camp per a futurs usos.
custodyURL	String	Si la petició ha definit custòdia, llavors aquest camp conté la URL al fitxer custodiat.

5.4.3.- Classe Sign

Representa una firma. Els camps d'aquesta classe són els següents:

Nom	Tipus	Descripció
iD	long	Identificador emprat pel sistema PortaFIB per identificar aquesta firma
serialNumber	BigInteger	Identificador del certificat emprat per firmar.
issuer	String	Informació de la persona extreta del certificat
subject	String	Informació del tipus de certificat o de l'autoritat certificadora que ha emès el certificat.

5.4.4.- Classe Actor

Els camps d'aquesta classe són els següents:

Nom	Tipus	Descripció
iD	String	Identificador de l'usuari-entitat dins PortaFIB.
name	String	Nom de la persona
administrationID	String	NIF de la persona

6.- Callback REST PortaFIB v1

Nou a v1.1.3

6.1.- Introducció

Callback REST funciona de forma similar a l'API de Callback WS, però fa servir serveis REST per a la comunicació amb les Aplicacions. Com hem dit abans, el Callback serveix al servidor PortaFIB per enviar notifikacions via REST dels events que ocorren en peticions relacionades amb els usuaris aplicació, és a dir, les peticions de firma enviades via WS emprant un usuari-aplicació. Aquest nou Servei s'ha implementat ja que és molt més senzill fer feina amb REST, ja que es pot incorporar directament en la capa WEB enlloc de fer tot un muntatge per escoltar peticions WS.

Únicament hem de configurar l'Usuari-Aplicació per a usar RESTS en les notifikacions i definir la ruta o URL on l'aplicació externa ha definit el servei REST.

Callback Versió (*)	PortaFIB Callback REST v1.0
Callback URL (*)	http://localhost:8080/portafib/cbrest/v1/event

En l'exemple anterior la ruta apunta al Servei Rest de proves que incorpora PortaFIB.

6.2.- Implementar servei REST en aplicació externa

A continuació es descriuen les passes per a que una aplicació externa implementi el servei REST i d'aquesta forma pugui rebre les notifikacions de les peticions que ha enviat a PortaFIB.

Podeu trobar un exemple del que s'explicarà en el codi font de PortaFIB en el directori [portafib-1.1]\ws\portafib_callback_server_rest.

6.2.1.- Dependències

Aquí teniu les dependències a afegir al pom.xml del mòdul Web:

```
<dependency>
  <artifactId>portafib-callback-beans</artifactId>
  <groupId>es.caib.portafib</groupId>
  <version>1.1.3</version>
</dependency>

<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-server</artifactId>
  <version>1.8</version>
</dependency>

<dependency>
```



```

<groupId>com.sun.jersey</groupId>
<artifactId>jersey-json</artifactId>
<version>1.8</version>
<exclusions>

    <exclusion>
        <groupId>stax</groupId>
        <artifactId>stax-api</artifactId>
    </exclusion>

    <exclusion>
        <groupId>com.sun.xml.bind</groupId>
        <artifactId>jaxb-impl</artifactId>
    </exclusion>

</exclusions>
</dependency>

```

6.2.2.- Controlador de Servei REST

Aquí teniu un exemple de “receptor” de peticions REST. En aquesta classe hem de tenir en compte el package (que es modificarà pel propi de l'aplicació) i la ruta o Path relatiu a aquest servei definit en la classe i en el mètode. En l'exemple següent són /v1 i /event, que conformen una ruta relativa /v1/event. Aquestes dues anotacions les hem de tenir en compte per gestionar els punts següents.

```

package es.caib.portafib.callback.rest.server.v1;

import javax.ws.rs.Consumes;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;
import es.caib.portafib.callback.beans.v1.PortaFIBEvent;
import es.caib.portafib.callback.beans.v1.testers.PortaFIBEventStore;

/**
 * @author anadal
 */
@Path("/v1")
public class PortaFIBRestServiceV1 {

    @POST
    @Path("/event")
    @Consumes(MediaType.APPLICATION_JSON)
    public Response event(PortaFIBEvent event) {

        try {
            // Fer el que sigui !!!!!
            return Response.status(200).entity("OK").build();
        } catch (Throwable th) {
            String msg = "Error desconegut processant event de Peticio de Firma REST: "
                + th.getMessage();
            th.printStackTrace();
            return Response.status(500).entity(msg).build();
        }
    }
}

```

Les classes i la gestió de les mateixes es descriuen en l'apartat de API Callback de WS.

6.2.3.- Modificacions en el web.xml

Ara toca associar el servlet de REST amb el controlador que hem programat en el punt anterior:

- (1) Propietat “com.sun.jersey.config.property.packages”: aquesta ha de constar de dos packages: el primer és el package d'on es troba el controlador i el segon sempre ha de ser “es.caib.portafib.callback.beans.v1”. S'han de separar per un ';’.
- (2) “Url-pattern”: Aquesta propietat indica la url base per tots els serveis definits en els controladors. En l'exemple de web.xml aquesta URL val /callbackrest. Si concatenem aquest valor amb el definit en el controlador, la ruta d'accés a aquest servei seria http://HOST:PORT/{aplicació_context}/callbackrest/v1/event, que és la que d'ha de posar en la definició de l'Usuari-Aplicació (Veure Introducció)

```
<servlet>
  <servlet-name>jersey-servlet</servlet-name>
  <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>com.sun.jersey.config.property.packages</param-name>
    <param-value>
es.caib.portafib.callback.rest.server.v1;es.caib.portafib.callback.beans.v1
    </param-value>
  </init-param>
  <init-param>
    <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>jersey-servlet</servlet-name>
  <url-pattern>/callbackrest/*</url-pattern>
</servlet-mapping>
```

7.- Annexes

7.1.- Classe UsuariAplicacioBean

Els camps de l'estructura UsuariAplicacioBean són els següents:

Nom	Tipus	Descripció
Actiu	boolean	Indicam si després de la creació, aquest estirà o no actiu. En consultes o llistats ens indica si està o no actiu en aquell moment.
CallbackURL	String	La forma que el PortaFIB per avisar d'esdeveniments que succeeixen durant el procés d'una firma és a través de cridades WebServices als client que han iniciat Peticions de Firma. En aquest camp s'ha de definir la URL a aquests serveis web de Callback per a que PortaFIB s'hi pugui connectar i enviar els avisos oportuns. Envia la informació d'estat de peticions mitjançant WebServices. Si volem emprar els servidors de prova de callback, llavors utilitzarem les següent URL segons la versió de callback elegida: <ul style="list-style-type: none">• Callback Versió 0: http://HOST:PORT/portafirmascb/web/services/MCGDWS• Callback Versió 1: http://HOST:PORT/portafibcb/v1/PortaFIBCallBack
CallbackVersio	int	Hi ha dues versions de WebServices per realitzar aquesta tasca: <ul style="list-style-type: none">(a) Versió 0: Utilitza l'API de callback de l'antic portafirmes de la CAIB. Només s'utilitza per retrocompatibilitat.(b) Versió 1: És una nova API de callback pròpia de PortaFIB escrita des de zero per adaptar-se a les necessitats de PortaFIB.
Contrasenya	String	És la contrasenya usada en el procés d'autenticació des d'un programa client. Per entorns CAIB pot valer qualsevol cosa excepte null.
Descripcio	String	(opcional): Descriu un poc la finalitat que tindrà aquest usuari-aplicació
EmailAdmin	String	És l'email on s'enviaran informacions o errors que involucrin aquest usuari-aplicació.
EntitatID	String	L'identificador de l'entitat a la que ha de pertànyer aquest usuari aplicació.

IdiomaID	String	En certes tasques internes d'un proces de firma iniciat per un usuari-aplicació, és necessita un idioma per traduir missatges i es aquest que elegirem aquí. Idioma en format "ISO 639-1". S'ha de comprovar que aquest idioma estigui suportat dins de l'entorn PortaFIB. El valors possibles són "ca" per català o "es" per castellà.
LogoSegell	FitxerBean	(opcional) És el Logo que s'utilitzarà en la taula de firmes duran l'adaptació del PDF al iniciar una Petició de Firma. Si no s'usa llavors s'utilitzarà el logo definit en l'entitat. L'objecte FitxerBean s'utilitza per enviar i rebre informació del logo (Veure punt "1.5.4.1.-Classe FitxerBean"). LogoSegellID és l'identificador intern del fitxer dins PortaFIB.
LogoSegellID	Long	
PotCustodiar	Boolean	Indica si aquest usuari-aplicació tindrà permisos per Custodiar. Valors possibles true (pot custodiar) i false o null (no pot custodiar)
UsuariAplicacioID	String	Nom Usuari: Aquest és el nom d'usuari o més conegut com <i>username</i> . Aquest és la cadena que s'utilitzarà per autenticar-se dins PortaFIB des dels programes que vulguin accedir al PortaFIB. En entorns no CAIB, aquest identificador ha de començar per l'identificador de l'entitat més el caràcter "_".

7.2.- Classe UsuariAplicacioFilterWs

Aquesta classe s'utilitza per filtrar UsuarisAplicacions sobre classes de tipus UsuariAplicacioBean (Veure "7.1.-Classe UsuariAplicacioBean")

Nota 1: Un camp null en qualsevol d'aquest camps significa que accepta qualsevol valor per aquell camp.

Nota 2: En el camps de tipus String, es poden aplicar caràcters comodí (o *wildcards*) com '%' que significa des de cap a múltiples caràcters o '_' que significa un simple caràcter. Per exemple "%nera" filtraria **prunera** i **llimonera**. La cadena de cerca '_o_a' filtraria **coca** i **sopa**.

Nom	Tipus	Descripció
FilterByActiu	Boolean	(opcional) Filtre pel camp actiu d'un Usuari-aplicació. El valors possibles són NULL, true o false.
FilterByCallBackURL	String	(opcional) Filtre que s'aplica sobre el camp callBackUrl de UsuariAplicacioBean
FilterByEntitatID	String	(opcional) Filtre que s'aplica sobre el camp entitatID de UsuariAplicacioBean
FilterByUsuariAplicacioID	String	(opcional) Filtre que s'aplica sobre el camp cusuariAplicacioID de UsuariAplicacioBean

7.3.- Classe UsuariPersonaBean

Classe que representa una Persona de PortaFIB.

Nom	Tipus	Descripció
Email	String	Correu electrònic de la Persona (no el personal sinó l'usat en l'entorn corporatiu o administratiu)
IdiomaID	String	Idioma en format "ISO 639-1". S'ha de comprovar que aquest idioma estigui suportat dins de l'entorn PortaFIB. El valors possibles són "ca" per català o "es" per castellà
Llinatges	String	Llinatges de la persona (o surname)
Nif	String	L'identificador com l'administració identifica a aquest individu o persona. Actualment és suporta el NIF
Nom	String	Nom de la persona (o given name)
Rubrica	FitxerBean	(opcional) Per a futures implementacions
RubricaID	Long	(opcional) Per a futures implementacions
UsuariPersonalID	String	És l'identificador emprat durant l'autorització i autenticació (o username)

7.4.- Classe UsuariEntitatBean

Classe que representa un Usuari-Entitat dins PortaFIB.

Nom	Tipus	Descripció
Actiu	boolean	Indicam si després de la creació, aquest estirà o no actiu. En consultes o llistats ens indica si està o no actiu en aquell moment
Carrec	String	No s'utilitza. Valdrà sempre null.
Email	String	(opcional) Ens indica un email específic per aquella persona dins l'entitat. Si val null llavors s'utilitza el email definit en l'Usuari-Persona
EntitatID	String	Identificador d'entitat amb la que volem enllaçar l'usuri-persona.
LogoSegell	FitxerBean	(opcional) És el Logo que s'utilitzarà en la taula de firmes duran l'adaptació del PDF al iniciar una Petició de Firma. Si no s'usa llavors s'utilitzarà el logo definit dins l'entitat. L'objecte FitxerBean s'utilitza per enviar i rebre informació del logo (Veure punt "1.5.4.1.-Classe FitxerBean"). LogoSegellID és l'identificador intern del fitxer dins PortaFIB.
LogoSegellID	Long	

PotCustodiar	Boolean	Indica si aquest usuari-aplicació tindrà permisos per Custodiar. Valors possibles true (pot custodiar) i false o null (no pot custodiar)
Predeterminat	boolean	Indica si aquesta entitat serà la predeterminada quan entri via web.
RebreTotsElsAvisos	boolean	Indica si volem rebre tots els avisos via email, independent de les Notificacions per Correu (Veure manual d'usuari) definides en l'entorn web.
UsuariEntitatID	String	En mètodes de creació aquest camp ha d'estar buit. En mètodes de consulta o llistat aquest camp contindrà l'identificador de l'Usuari-Entitat
UsuariPersonalID	String	Identificador de l'usuari persona que volen associar a l'entitat definida en el camp EntitatID.

7.5.- Classe CarrecWs

Classe que s'utilitza per la gestió de càrrecs dins d'una entitat.

Nom	Tipus	Descripció
Actiu	boolean	Indica si aquest carrec està o no actiu.
CarrecID	String	Identificador de càrrec.
CarrecName	String	Nom del Càrrec
EntitatID	String	Entitat a la que pertany el càrrec
UsuariPersonalID	String	Identificador d'usuari-persona de la persona que ostenta el càrrec.

7.6.- Classe FluxDeFirmesWs

Classe que representa les persones que firmaran el document apart de l'ordre i les condicions en les que ho faran.

Camp	Tipus	Descripció
FluxDeFirmesID	long	Identificador del Flux de Firmes al que pertany aquest bloc. Durant les operacions de creació valdrà null.
nom	String	Nom que li donarem a aquest flux de firmes.
BlocsDeFirmes	Set<BlocDeFirmesWs>	Conjunt de blocs que formen aquest flux de firmes. Els detalls de la classe BlocDeFirmesBean es poden consultar al punt "7.7.-Classe BlocDeFirmesWs"

7.7.- Classe BlocDeFirmesWs

Representa a un conjunt de firmes dins d'un Flux de Firmes.

Camp	Tipus	Descripció
BlocDeFirmesID	long	Identificador del BlocDeFirmes. Durant les operacions de creació valdrà null.
Ordre	int	A causa de que els bloc és guarden dins un conjunt de dades no ordenat requerim saber en quin ordre hem d'anar executant les firmes de cada bloc.
DataFinalitzacio	Timestamp	Ha de valer null. Conté una data a partir del moment en que totes les firmes d'aquell bloc han finalitzat (firmat, rebutjat o descartat)
MinimDeFirmes	int	És el conjunt mínim de firmes requerides per passar al següent bloc. El valor mínim serà la suma de firmes obligatòries del camp Firmes i el valor màxim serà la suma de firmes obligatòries més 1 en el cas d'haver-hi firmes opcional.
FluxDeFirmesID	long	Identificador del Flux de Firmes al que pertany aquest bloc
Firmes	Set<FirmaBean>	Conjunt de firmes incloses dins aquest bloc. Els detalls de la classe FirmaBean es poden consultar al punt "7.8.- Classe FirmaBean".

7.8.- Classe FirmaBean

Classe que representa la firma d'una persona tant si és la planificació (flux de firma) com l'execució (el document firmat).

Camp	Tipus	Descripció
BlocDeFirmaID	long	Identificador del BlocDeFirmes al que pertany aquesta firma. Durant les operacions de creació valdrà null.
CaixaAlt	Integer	NO IMPLEMENTAT
CaixaAmple	Integer	NO IMPLEMENTAT
CaixaPagina	int	NO IMPLEMENTAT
CaixaX	Integer	NO IMPLEMENTAT
CaixaY	Integer	NO IMPLEMENTAT
DestinatariID	String	Identificador de l'usuari entitat que requerim que firmi. Aquí també podem definir identificadors de Càrrecs.
EmissorCertificat	String	Una vegada aquesta persona hagi firmat, aquest camp s'omplirà amb informació de l'emissor del certificat utilitzat per la firma.
FirmaID	long	Identificador de la Firma. Durant les operacions de creació valdrà null.
FitxerFirmat	FitxerBean	Una vegada el DestinatariID hagi firmat aquest camp

FitxerFirmatID	long	contindrà el fitxer firmat.
MostrarRubrica	boolean	NO IMPLEMENTAT
NomCertificat	String	Una vegada aquesta persona hagi firmat, aquest camp s'omplirà amb el nom definit en el certificat amb el que s'ha firmat el document.
NumFirmaDocument	Integer	Una vegada firmat conté el numero de firma, és a dir, si és la tercera persona que firma el document, llavors aquest valdrà 3.
NumeroSerieCertificat	BigInteger	Una vegada aquesta persona hagi firmat, aquest camp s'omplirà amb el número de sèrie del certificat emprant per firmar.
Obligatori	boolean	Indica si aquesta firma és obligatòria. És a dir. La firma pot estar dins un bloc on dues o més persones són requerides per realitzar una sola firma, en aquest cas, la firma no és obligatòria i aquest camp val "false". Està molt relacionat amb el camp "MinimDeFirmes" de la classe BlocDeFirmes (Veure punt 7.7.-Classe BlocDeFirmesWs)
TipusEstatDeFirmaFinalID	Long	<p>Indica els estats en que es pot trobar una firma:</p> <ul style="list-style-type: none"> • NULL: indica que la firma no s'ha realitzat. • TIPUSESTATDEFIRMAFINAL_FIRMAT = 2: Indica que s'ha firmat correctament. • TIPUSESTATDEFIRMAFINAL_REBUTJAT = 3: Quan en aquesta firma s'ha decidit rebutjar la petició de firma. • TIPUSESTATDEFIRMAFINAL_DESCARTAT=4: Es passa a aquest estat quan la tasca encomanada ha sigut realitzada per un altra persona (Cas de firmes no obligatòries) <p>Durant les operacions de creació aquest camp valdrà null. Aquestes constants estan definides dins la classe <code>es.caib.portafib.utils.Constants</code>.</p>

7.9.- Classe CustodialInfoBean


Classe que serveix per informar a PortaFIB que una vegada el document s'hagi firmat, llavors aquest s'ha de custodiar.


Camp	Tipus	Descripció
codiBarresID	String	Identificador del tipus de codi de barres a utilitzar durant l'estampació de la informació de custòdia. Al final d'aquesta taula es mostren els possibles valors per aquest camp.


codiBarresPosicioPaginaID	long	NO IMPLEMENTAT. Aquest camp ha de tenir el mateix valor que el camp "MissatgePosicioPaginaID" d'aquesta classe
codiBarresText	String	Text de custòdia a integrar amb el codi de barres. En el missatge es poden fer les següents substitucions: <ul style="list-style-type: none"> • {0} = URL de validació • {1} = custodialID • {2} = custodiaPluginClassID • {3} = data amb hora • {4} = valor especial Normalment s'hauria d'utilitzar "{0}"
custodiaInfoID	long	Identificador d'aquesta custòdia dins PortaFIB. Durant les operacions de creació valdrà null.
custodiaDocumentID	String	Contindrà l'identificador emprat en el propi sistema de custòdia. Valor assignat pel servidor.
custodiaPluginClassID	String	Identificador del Plugin que gestionarà la Custòdia. Actualment hi ha dues implementacions: <ul style="list-style-type: none"> • Sistema de Fitxers. • Custòdia CAIB. De totes formes aquest valor s'ha d'obtenir cridant al mètode getCurrentCustodiaClass() de l'API de Peticio de Firma (Veure punt "4.3.4.-getCurrentCustodiaPluginClass()")
custodiaPluginID	String	Identificador retornat pel sistema de custòdia.
custodiaPluginParameters	String	Paràmetres addicionals requerits pel sistema de custòdia.
custodiar	Boolean	Serveix per desactivar la custòdia del document firmat, però sense perdre la configuració de custòdia.
dataCustodia	Timestamp	Data de la custòdia del document firmat dins el sistema de custòdia.
editable	Boolean	Ens indica si la informació de Custòdia pot ser modificada o no. Si val false, llavors s'ha d'obtenir del servidor i no pot ser modificada.
entitatID	String	NO IMPLEMENTAT. Ha de valer null.
missatge	String	Missatge de custòdia a mostrar en el document. Es poden fer les següents substitucions: <ul style="list-style-type: none"> • {0} = URL de validació • {1} = custodialID • {2} = custodiaPluginClassID • {3} = data amb hora • {4} = valor especial

missatgePosicioPaginaID	long	Indica el lateral de la pàgina on es situarà el missatge i el codi de barres. Els valors possibles són: <ul style="list-style-type: none"> • Constants.POSICIO_PAGINA_CAP=0; • Constants.POSICIO_PAGINA_ADALT = 1; • Constants.POSICIO_PAGINA_ABAIX = 2; • Constants.POSICIO_PAGINA_ESQUERRA = 3 • Constants.POSICIO_PAGINA_DRETA = 4;
nomPlantilla	String	NO IMPLEMENTAT. Ha de valer null.
pagines	String	Describeix les pàgines on s'estamparà la informació de custòdia. Els valors possibles són: <ul style="list-style-type: none"> • buit = no mostrar en cap pàgina • * = totes les pàgines • 0 = primera pàgina (taula de firmes) • -1 = darrera pàgina (taula de firmes) • X-Y o X- = Rang de pàgines. Indica de la pàgina X a la Y o de la pàgina X fins al final respectivament. • -1, 0, 1, 3, 4-5, 8- = Combinació dels formats anteriors separats per una coma
pluginID	long	Identificador del Plugin de Custòdia. Aquests valor el definirà el Servidor.
titolPeticio	String	Text descriptiu del document a custodiar. Normalment s'indica el títol de la petició de Firma a la que pertany.
urlFitxerCustodiat	String	Url de validació del document retornada pel sistema de custòdia. Durant la creació ha de valer null.
usuariAplicacioID	String	Ha de valer l'username de l'usuari aplicació que realitza la petició webservice.
usuariEntitatID	String	NO IMPLEMENTAT. Ha de valer null.

Actualment, a part d'altres possibles implementacions de generadors de codis de barres, els identificadors de codis de disponibles són:

Codi de Barres PDF417	
Variable:	Constants.BARCODE_PDF417_PLUGIN
Valor:	org.fundaciobit.plugins.barcode.pdf417.Pdf417Plugin
Exemple:	

Codi de Barres QR	
Variable:	Constants.BARCODE_QR_PLUGIN
Valor:	org.fundaciobit.plugins.barcode.qrcode.QrCodePlugin
Exemple:	

Codi de Barres 128	
Variable:	Constants.BARCODE_128_PLUGIN
Valor:	org.fundaciobit.plugins.barcode.barcode128.BarCode128Plugin
Exemple:	

7.10.- Classe TipusDocumentInfoWs

Classe que encapsula la informació bàsica d'un tipus de document.

Camp	Tipus	Descripció
Nom	String	Nom del tipus de document
TipusDocumentID	long	Identificador del tipus de document

7.11.- Classe MetadadaBean

Actualment no s'utilitza per res. En un futur servirà per afegir Metadades a la Petició de Firma.

Camps	Tipus	Descripció
Descripcio	String	(opcional) Descripció de la metadada
MetadadaID	long	Identificador de la metadada. Durant les operacions de creació valdrà 0.
Nom	String	Nom de la metadada
PeticioDeFirmaID	long	Identificador de la Petició de Firma amb la que esta relacionat aquest Annex. Durant les operacions de creació valdrà 0.

TipusMetadadaID	int	Indica el tipus de dada que conté el camp Valor. El valors possibles són: <ul style="list-style-type: none"> Constants.TIPUSMETADADA_STRING = 0; Constants.TIPUSMETADADA_INTEGER = 1; Constants.TIPUSMETADADA_DECIMAL = 2; Constants.TIPUSMETADADA_BOOLEAN = 3; Constants.TIPUSMETADADA_BASE64 = 4; Constants.TIPUSMETADADA_DATE = 5; Al final d'aquesta taula trobareu una altra taula on es descriure les classes utilitzades per convertir de String a tipus descrit aquí i a l'inrevés.
Valor	String	El valor de la metadada en format cadena de caràcters.

A continuació es mostra la taula amb les Classes java utilitzades per la conversió a String o des de String segons el camp "TipusMetadadaID" de la classe MetadadaBean.

Tipus	Classe
TIPUSMETADADA_STRING	java.lang.String
TIPUSMETADADA_INTEGER	java.math.BigInteger
TIPUSMETADADA_DECIMAL	java.math.BigDecimal
TIPUSMETADADA_BOOLEAN	java.lang.Boolean
TIPUSMETADADA_BASE64	org.fundaciobit.plugins.utils.Base64 que pertany al projecte PluginsIB de la Fundació Bit (Cercar-ho a sourceforge)
TIPUSMETADADA_DATE	org.fundaciobit.plugins.utils.ISO8601 que pertany al projecte PluginsIB de la Fundació Bit (Cercar-ho a sourceforge)

7.12.- Classe AnnexBean

Representa la informació d'un Annex d'una Petició de Firma.

Camp	Tipus	Descripció
Adjuntar	boolean	Indica si aquest fitxer s'ha d'adjuntar al document de firma de la petició de firma. Pot valer true o false en el cas de que el tipus de firma sigui PADES. En altres casos sempre valdrà false.
AnnexID	long	Identificador de l'annex. Durant les operacions de creació valdrà 0.

Firmar	boolean	Indica si l'annex s'ha de firmar. Actualment només es permet un valor true si el camp adjuntar es true i el tipus de firma de la petició de firma es PADES
Fitxer	long	Dades que representen l'annex.
FitxerID	FitxerBean	
PeticioDeFirmaID	long	Identificador de la Petició de Firma amb la que esta relacionat aquest Annex. Durant les operacions de creació valdrà 0.

7.13.- Classe PeticioDeFirmaWs

Classe que representa tota la informació d'una petició de firma.

Camp	Tipus	Descripció
PeticioDeFirmaID	long	Identificador de la Petició de Firma. Durant les operacions de creació valdrà 0.
Títol	String	Títol de la Petició de Firma
Descripcio	String	Descripció de la Peticio de Firma
Motiu	String	Text orientat als destinataris indicant la raó de perquè se'ls hi envia aquest document per a que el signin.
FitxerAFirmarID	long	Fitxer a firmar. Veure punt "1.5.4.1.-Classe FitxerBean"
FitxerAFirmar	FitxerBean	
FitxerAdaptatID	Long	Durant les operacions de creació valdrà null. Només per tipus de firma PADES. És el fitxer a firmar però amb les següents modificacions:
FitxerAdaptat	FitxerBean	<ul style="list-style-type: none"> • Annexes adjunts • Taula de Firmes • Estampació de informació de Custòdia
TipusDocumentID	long	Identificador del tipus de document que conté el camp "FitxerAFirmar". Podeu consultar tots els tipus disponibles emprant el mètode <code>getTipusDeDocuments</code> (Veure punt "4.3.6.-getTipusDeDocuments()")
DescripcioTipusDocument	String	(opcional) En cas de que el tipus de document sigui Altres o "Otros", llavors en aquest camp poden descriure el tipus de document.

PosicioTaulaFirmesID	int	Només per tipus de firma PADES. Indica si s'afegirà una taula de firmes en el document PDF. El valors possibles són: <ul style="list-style-type: none"> • TAULADEFIRMES_SENSETAULA = 0 • TAULADEFIRMES_PRIMERAPAGINA = 1 • TAULADEFIRMES_DARRERAPAGINA = -1
DataSolicitud	Timestamp	Ho omple el sistema amb la data en que es va iniciar la petició.
DataFinal	Timestamp	Ho omple el sistema amb la data en que es va realitzar la darrera firma del document o la data en que es va rebutjar. Durant les operacions de creació valdrà null.
DataCaducitat	Timestamp	Estableix una data aproximada de quan aquesta petició hauria d'haver acabat.
TipusFirmaID	int	Defineix el tipus de firma a realitzar: <ul style="list-style-type: none"> • Constants.TIPUSFIRMA_PADES = 0 • Constants.TIPUSFIRMA_XADES = 1 • Constants.TIPUSFIRMA_CADES = 2 En aquests moments únicament es suporta el tipus PADES (0).
AlgorismeDeFirmaID	long	Algorisme que s'utilitzarà per realitzar la firma. El valor per aquest camp ha de ser 0 (SHA1WITHRSA)
ModeDeFirma	Boolean	Ha de valer false indicant el mode Implicit. True per mode explícit.
TipusEstatPeticioDeFirmaID	int	Indica l'estat de la petició. Ho modifica el sistema PortaFIB. Per veure el significat dels valors d'aquest cap consultar el mètode "4.3.12.-getStateOfPeticioDeFirma()"
MotiuDeRebuig	String	Durant les operacions de creació valdrà null. Ho omple el sistema amb el motiu de rebuig quan algun destinatari o delegat rebutja la petició.
IdiomaID	String	Idioma en format "ISO 639-1". S'ha de comprovar que aquest idioma estigui suportat dins de l'entorn PortaFIB. El valors possibles són "ca" per català o "es" per castellà. S'utilitzarà principalment pels missatges a estampar en el document.
PrioritatID	int	Indica la prioritat de la petició. Els valors possibles són: <ul style="list-style-type: none"> • Constants.PRIORITAT_BAIXA = 0 • Constants.PRIORITAT_NORMAL = 5 • Constants.PRIORITAT_ALTA = 9
UsuariAplicacioID	String	Ho omple el sistema amb l'username de l'usuari aplicació que realitza la petició.

RemitentNom	String	El nom de qui envia la petició de firma. Normalment es posa l'aplicació d'on ve la petició (Helium, Sistra, ...)
RemitentDescripcio	String	Detalls del remitent, normalment es posa un email de contacte de la persona encarregada de l'aplicació que envia la petició.
InformacioAdicional	String	Informació que l'usuari aplicació
LogoSegellID	Long	NO IMPLEMENTAT. Valdrà NULL
UsuariEntitatID	String	Ha de valer null. Utilitzat per peticions de firma de persones.
AvisWeb	boolean	Ha de valer false. Utilitzat per peticions de firma de persones.
Annexs	Set<AnnexBean>	(opcional) Llistat d'annexes associats a la petició de firma. Veure punt "7.12.-Classe AnnexBean"
CustodialInfoID	Long	(opcional) Informació de Custòdia si el servidor ho suporta. Veure punt "7.9.-Classe CustodialInfoBean". Durant les creacions CustodialInfoID ha de valer null.
CustodialInfo	CustodialInfoBean	
FluxDeFirmesID	long	Informació de les persones que han de firmar el document de la petició de firma. Durant les creacions FluxDeFirmesID ha de valer null. Veure punt "7.6.-Classe FluxDeFirmesWs"
Flux De Firmes	FluxDeFirmesWs	
Metadades	Set<Metadada>	NO IMPLEMENTAT.