



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

una manera de hacer
europa 

Fondo Europeo de
Desarrollo Regional



Unió Europea

NOTIB: Manual d'integracions

28 de setembre de 2018

Serveis d'Administració Electrònica en el Govern de les Illes Balears

Lot 3 (Serveis de tramitació d'expedients electrònics)

Oficina Tècnica de Direcció de Projecte

Control de versions del document

Control de Canvis			
Data	Autor	Versió	Canvis
31/08/2018	Limit Technologies	v1.0	Versió inicial

Revisat per		
Nom	Data	Àrea, departament o empresa

Aprovat per		
Nom	Data	Àrea, departament o empresa

Llista de distribució		
Nom	Àrea, departament o empresa	Correu electrònic

Índex

Control de versions del document.....	2
1. Introducció.....	4
2. Model de dades.....	4
3. Operacions de notificació.....	9
3.1. Operació altaRegistroEntrada.....	10
3.2. Operació consultaEstatNotificacio.....	10
3.3. Operacio consutaEstatEnviament.....	10
3.4. Callback.....	11
4. Client.....	11
4.1. Importació de la llibreria del client.....	11
4.2. Ús de la llibreria.....	12
4.3. Exemple ús llibreria.....	13

1. Introducció

En aquest manual es detallen els serveis que ofereix la aplicació Notib per a que altres aplicacions s'hi puguin integrar per tal de poder donar d'alta notificacions, i consultar-les.

S'han definit els serveis amb el protocol REST. La seva URL d'accés és:

```
http(s)://<NOTIB_HOST>/notib/api/services
```

També s'ha definit una URL per a poder consultar els serveis actualment existents. Aquesta URL és:

```
http(s)://<NOTIB_HOST>/notib/api/apidoc
```

En aquesta adreça es visualitzarà una pàgina amb totes les operacions disponibles, informant dels seu model de dades, i amb exemples.

api/services/notificacio			Mostrar/Ocultar	Llistar Operacions	Expandir Operacions
POST	/api/services/notificacio/alta	Genera una notificació			
GET	/api/services/notificacio/consultaEstatEnviament/{referencia}	Consulta de la informació d'una notificació			
GET	/api/services/notificacio/consultaEstatNotificacio/{referencia}	Consulta de la informació d'una notificació			
[BASE URL: /notib]					

2. Model de dades

A continuació es descriuen els models de dades que gestiona l'aplicació NOTIB amb les interfícies de serveis REST, que permeten gestionar les notificacions de terceres aplicacions.

Definició d'objecte: Notificacio			
Camp	Tipus	Car	Descripció
emisorDir3Codi	cadena	1	Codi Dir3 de l'organisme emisor
comunicacioTipus	cadena		Enumerat que indica si al donar d'alta una notificació a Notib també s'ha de donar d'alta a Notific@ en la mateixa operació, o si s'ha de fer amb un procés en segon pla posteriorment. Valors: [SINCRON, ASINCRON]
enviamentTipus	cadena	1	Enumerat que indica si l'enviament és una comunicació o una notificació. Valors: [NOTIFICACIO, COMUNICACIO]
concepte	cadena	1	Concepte de l'enviament. Si ha d'anar a CIE només s'agafaran els 50 primers caràcters.
descripcio	cadena	0..1	Descripció detallada de l'enviament

enviamentDataProgramada	data	0..1	Data en la que l'enviament es posarà a disposició per a la compareixença
retard	enter	0..1	Dies que l'enviament estarà a disposició de compareixença en carpeta abans d'entregar-lo per altres mitjans
caducitat	data	0..1	Data d'expiració de l'enviament
document	Document	1	Document que s'envia en la notificació
procedimentCodi	cadena	1	Identificador del procediment SIA al que pertany la notificació
pagadorPostal	PagadorPostal	0..1	Dades de l'organisme que assumeix els costos postals
pagadorCie	PagadorCie	0..1	Dades de l'organisme que assumeix els costos del CIE
enviaments	Enviament	1..n	Enviaments continguts en la
parametresSeu	ParametresSeu	1	Paràmetres necessaris per a poder fer l'enviament de la notificació a la Seu de la Caib

Definició d'objecte: Document			
Camp	Tipus	Car	Descripció
arxiuNom	cadena	1	Nom de l'arxiu
contingutBase64	cadena	0..1	Contingut del document en base 64. Obligatori si no està informat l'enllaç extern. * En la v.0.1 és obligatori.
hash	cadena	1	Hash del document. El hash es va amb sha256 i es codifica en base64
url	cadena	0..1	Enllaç extern on es troba el document de l'enviament. Obligatori si no està informat el contingut del document.
metadades	cadena	0..1	Paràmetres extres del propi document en format XML.
notmalitzat	booleà	0..1	Indica si el document està normalitzat per a la impressió al CIE. Només aplica per enviaments postals.
generarCsv	booleà		Indica si es desitja que Notific@ substitueixi el document per un altre en el que s'inclogui un Codi Segur de Verificació. Només aplica amb PDF amb firma PADES.

Definició d'objecte: PagadorPostal			
Camp	Tipus	Car	Descripció
dir3Codi	cadena	1	Codi Dir3 de l'organisme pagador dels costos postals
contracteNum	cadena	1	Número del contracte amb l'operador postal
contracteDataVigencia	data	1	Data de vigència del contracte amb l'operador postal
facturacioClientCodi	cadena	1	Codi de client de facturació amb l'operador postal

Definició d'objecte: PagadorCie			
Camp	Tipus	Car	Descripció
dir3Codi	cadena	1	Codi Dir3 de l'organisme pagador dels costos de impressió i ensobrat
contracteDataVigencia	data	1	Data de vigència del contracte amb el CIE

Definició d'objecte: ParametresSeu			
Camp	Tipus	Car	Descripció
expedientSerieDocumental	cadena	0..1	Sèrie documental de l'expedient al que pertany la notificació
expedientUnitatOrganitzativa	cadena	1	Unitat organitzativa SISTRA de l'expedient al que pertany la notificació
expedientIdentificadorEni	cadena	1	Identificador de l'expedient en format ENI. Format: <i>ES_<Órgano>_<AAAA>_<ID_específico></i> *Longitud total: 50 caràcters
expedientTitol	cadena	1	Títol de l'expedient (Caràcters vàlids: [AZaz-/])
procedimentCodi	cadena	1	Codi del procediment SISTRA de l'expedient al que pertany la notificació
registreOrgan	cadena	1	Òrgan on registrar la notificació SISTRA
registreOficina	cadena	1	Oficina on registrar la notificació SISTRA
registreLlibre	cadena	1	Llibre on registrar la notificació SISTRA
idioma	cadena	1	Idioma de la notificació. Valors possibles: [ca, es]
avisTitol	cadena	1	Títol de l'avis de notificació al ciutadà
avisText	cadena	1	Text descriptiu de l'avis
avisTextMobil	cadena	0..1	Text personalitzat per a l'avis per SMS
oficiTitol	cadena	1	Títol de l'ofici de remissió que es mostra una vegada oberta la notificació
oficiText	cadena	1	Text descriptiu de l'ofici

Definició d'objecte: Enviament			
Camp	Tipus	Car	Descripció
referencia	cadena	0..1	Referència única que estableix l'organisme emisor per a l'enviament. Si no s'indica Notib en generarà una automàticament.
titular	Persona	1	Titular de la notificació
destinatari	Persona	0..n	Conjunt de destinataris de la notificació
entregaPostal	EntregaPostal	0..1	Dades per a l'entrega postal
entregaDeh	EntregaDeh	0..1	Dades per a l'entrega a la DEH
serveiTipus	cadena	1	Enumerat que indica el tipus de servei de l'enviament. Valors possibles: [NORMAL, URGENT]

Definició d'objecte: Persona			
Camp	Tipus	Car	Descripció
nom	cadena	1	Nom de la persona física, o raó social de persona jurídica.
l·linatge1	cadena	0	Primer l·linatge de la persona física. *Obligatori en cas de persones físiques.
l·linatge2	cadena	0	Segon l·linatge de la persona física. *Obligatori en cas de persones físiques.
nif	cadena	1	NIF de la persona física o jurídica

telefon	cadena	0	Telèfon de la persona. *Actualment no s'utilitza
email	cadena	0	Correu electrònic de la persona. Àltament recomanat el seu ús.

Definició d'objecte: EntregaPostal			
Camp	Tipus	Car	Descripció
tipus	cadena	1	Enumerat que indica el tipus de entrega postal. Valors possibles: [NACIONAL, ESTRANGER, APARTAT_CORREUS, SENSE_NORMALITZAR]
viaTipus	cadena	0..1	Enumerat que indica el tipus de via per a entregues postals. Valors possibles: [ALAMEDA, CALLE, CAMINO, CARRER, CARRETERA, GLORIETA, KALEA, PASAJE, PASEO, PLAÇA, PLAZA, RAMBLA, RONDA, RUA, SECTOR, TRAVESIA, URBANIZACION, AVENIDA, AVINGUDA, BARRIO, CALLEJA, CAMI, CAMPO, CARRERA, CUESTA, EDIFICIO, ENPARANTZA, ESTRADA, JARDINES, JARDINS, PARQUE, PASSEIG, PRAZA, PLAZUELA, PLACETA, POBLADO, VIA, TRAVESSERA, PASSATGE, BULEVAR, POLIGONO, OTROS] * Obligatori quan tipoDomicili = NACIONAL
viaNom	cadena	0..1	Nom de la via per entregues postals * Obligatori quan tipoDomicili = NACIONAL o ESTRANGER
numeroCasa	cadena	0..1	Número de la casa per entregues postals * Obligatori quan tipoDomicili = NACIONAL excepte si s'indica el punt quilomètric
numeroQualificador	cadena	0..1	Informació addicional sobre la numeració de l'adreça. P. ex. «bis»
puntKm	cadena	0..1	Punt quilomètric per entregues postals * Obligatori quan tipoDomicili = NACIONAL excepte si s'indica numeroCasa
apartatCorreus	cadena	0..1	Apartat de correus per entregues postals * Obligatori quan tipoDomicili = APARTAT_CORREUS
portal	cadena	0..1	Portal de la casa per entregues postals
escala	cadena	0..1	Escala de la casa per entregues postals
planta	cadena	0..1	Planta de la casa per entregues postals
porta	cadena	0..1	Porta de la casa per entregues postals
bloc	cadena	0..1	Bloc de la casa per entregues postals
complement	cadena	0..1	Informació extra sobre la casa per entregues postals
codiPostal	cadena	1	Codi postal de la casa per entregues postals. Per enviaments internacionals a països sense codi posatal es pot posar 00000.
poblacio	cadena	0..1	Població on s'entrega l'enviament * Obligatori quan tipoDomicili = NACIONAL, ESTRANGER o APARTAT_POSTAL
municipiCodi	cadena	0..1	Codi INE del municipi on s'entrega l'enviament * Obligatori quan tipoDomicili = NACIONAL o APARTAT_POSTAL
provinciaCodi	cadena	0..1	Codi de la província on s'entrega l'enviament * Obligatori quan tipoDomicili = NACIONAL o APARTAT_POSTAL
paisCodi	cadena	0..1	Codi ISO del país on s'entrega l'enviament * Obligatori quan tipoDomicili = ESTRANGER

linia1	cadena	0..1	Línia 1 de l'adreça d'entrega de l'enviament sense normalitzar * Obligatori quan tipoDomicili = SENSE_NORMALITZAR
linia2	cadena	0..1	Línia 2 de l'adreça d'entrega de l'enviament sense normalitzar * Obligatori quan tipoDomicili = SENSE_NORMALITZAR
cie	cadena	0..1	Identificador del CIE en Notific@ per a seleccionar-lo per a l'enviament
formatSobre	cadena	0..1	Cadena indicant el format del sobre (americano, C5, ...)
formatFulla	cadena	0..1	Cadena indicant el format de la fulla (A4, A5, ...)

Definició d'objecte: EntregaDeh

Camp	Tipus	Car	Descripció
obligat	booleà	1	Indica si l'enviament és obligatori o voluntari
procedimentCodi	cadena	0..1	Codi del procediment en la DEH per a enviaments voluntaris

Definició d'objecte: RespostaAlta

Camp	Tipus	Car	Descripció
identificador	cadena	1	Identificador de la notificació a Notib
estat	cadena	1	Estat de la notificació a Notib
referencies	EnviamentReferencia	1..n	Referències dels enviaments de la notificació a Notib
error	booleà	0..1	Indica si s'ha produït algun error
errorDescripcio	cadena	0..1	Descripció de l'error

Definició d'objecte: EnviamentReferencia

Camp	Tipus	Car	Descripció
titularNif	cadena	1	Nif de la persona a qui va dirigit l'enviament
referencia	cadena	1	Identificador de l'enviament a Notib

Definició d'objecte: RespostaConsultaEstatNotificacio

Camp	Tipus	Car	Descripció
estat	cadena	1	Estat de la notificació a Notib
error	booleà	0..1	Indica si s'ha produït algun error en la notificació
errorData	data	0..1	Data de l'error
errorDescripcio	cadena	0..1	Descripció de l'error

Definició d'objecte: RespostaConsultaEstatEnviament

Camp	Tipus	Car	Descripció
estat	cadena	1	Estat de l'enviament a Notib
estatData	data	1	Data en que es va assignar l'estat a l'enviament
estatDescripcio	cadena	1	Descripció de l'estat de l'enviament
estatOrigen	cadena	0	

receptorNif	cadena	1	NIF del receptor de l'enviament
receptorNom	cadena	1	Nom del receptor de l'enviament
certificacio	Certificacio	0..1	Certificació generada al recepcionar la notificació. Aplicable únicament si es tracta d'una notificació, i no una comunicació, i si ha estat recepcionada.
error	booleà	0..1	Indica si s'ha produït algun error en l'enviament
errorData	data	0..1	Data de l'error
errorDescripcio	cadena	0..1	Descripció de l'error.

Definició d'objecte: Certificacio			
Camp	Tipus	Car	Descripció
data	data	1	Data de la certificació
origen	cadena	1	Origen de la certificació: - correos_cie (Postal) - electronico (DEH) - carpeta - sede - ema - funcionario_habilitado
contingutBase64	cadena	1	Contingut del document de la certificació de l'enviament
tamany	enter	1	Mida del document en bytes
hash	cadena	1	Hash del document
metadades	cadena	0..1	Metadades extres del propi document
csv	cadena	0..1	CSV del document
tipusMime	cadena	1	Tipus MIME del document

3. Operacions de notificació

Les operacions exposades per NOTIB, per tal de poder integrar-s'hi són les següents:

- **alta:** permet donar d'alta una notificació al sistema.
- **consultaEstatNotificacio:** permet obtenir l'estat de la notificació.
- **consultaEstatEnviament:** permet obtenir l'estat d'un enviament, i en cas que es trobi notificat el corresponent justificant de notificació.

3.1. Operació altaRegistroEntrada

Aquesta operació permet donar d'alta una notificació al sistema. Aquesta notificació es tramitarà mitjançant l'aplicació Notific@ i a la Seu de la Caib.

Operació: alta			
PETICIÓ			
Camp	Tipus	Car	Descripció
notificacio	Notificacio	1	Objecte amb tota la informació necessària per a poder crear una nova notificació en el sistema, i enviar-la a Notific@ i a la Seu de la Caib
RESPOSTA			
Camp	Tipus	Car	Descripció
resposta	RespostaAlta	1	

3.2. Operació consultaEstatNotificacio

Aquesta operació permet obtenir l'estat de la notificació en el sistema.

Operació: consultaEstatNotificacio			
PETICIÓ			
Camp	Tipus	Car	Descripció
identificador	cadena	1	Identificador de la notificació
RESPOSTA			
Camp	Tipus	Car	Descripció
resposta	RespostaConsultaEstatNotificacio	1	Contindrà l'estat actual de la notificació

3.3. Operació consutaEstatEnviament

Aquesta operació permet obtenir l'estat d'un enviament, i en cas que es trobi notificat el corresponent justificant de notificació.

Operació: consutaEstatEnviament			
PETICIÓ			
Camp	Tipus	Car	Descripció
identificador	Notificacio	1	Identificador de l'enviament
RESPOSTA			
Camp	Tipus	Car	Descripció
resposta	RespostaConsultaEstatEnviament	1	Contindrà la informació de l'estat de la notificació, del receptor de la notificació, i, si la notificació ha estat recepcionada, el corresponent justificant de recepció

3.4. Callback

A part de les operacions implementades per Notib, aquest disposa d'un servei per a avisar a les aplicacions tercers de que una notificació ha sofert algun canvi en el seu estat.

Aquest servei realitza una crida http a l'adreça que l'usuari d'aplicació tingui configurada a Notib.

Concretament a Notib es configura la URL allà on hi haurà d'haver configurades totes les operacions de callback, tot i que actualment únicament n'hi ha una d'implementada.

La URL de la operació ha de ser:

http://<Path_to_app_in_Notib>/notificaCanvi

Operació: notificaCanvi			
PETICIÓ			
Camp	Tipus	Car	Descripció
notificacioCanvi	NotificacioCanviClient	1	Informació de la notificació
RESPOSTA			
Sense resposta			

Definició d'objecte: notificacioCanviClient			
Camp	Tipus	Car	Descripció
identificador	cadena	1	Identificador de l'enviament actualitzat
referenciaEnviament	cadena	1	Referència de l'enviament actualitzat
data	data	1	Data de la modificació de l'enviament

4. Client

Per tal de simplificar la integració de aplicacions de tercers, desenvolupades en Java, amb Notib, s'ha desenvolupat una llibreria que inclou totes les classes necessàries per a l'intercanvi d'informació, i per a la realització de les peticions.

4.1. Importació de la llibreria del client

Aquesta llibreria es pot incloure en el projecte de diverses maneres:

- Copiant el fitxer jar de la llibreria al projecte.
 La llibreria es pot trobar en el propi projecte Notib, a la ruta:
notib/notib-client/target/notib-client-{versio}.jar, allà on {versio} és el nombre de versió de Notib amb el que es integrar la aplicació de tercers.

- Utilitzant Maven. Per incloure la llibreria a un projecte maven és necessari incloure la dependència següent:

```

<dependency>
  <groupId>es.caib.notib</groupId>
  <artifactId>notib-client</artifactId>
  <version>${notib.version}</version>
</dependency>

```

i el repositori de govern

```

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

```

4.2. Ús de la llibreria

Per a utilitzar la llibreria en una aplicació, un cop importada la llibreria al projecte, només cal inicialitzar el client:

```

private NotificacioService notificacioService;
private NotificacioService getNotificacioService() {
  if (notificacioService == null) {
    notificacioService = NotificacioRestClientFactory.getRestClient(
      URL_NOTIB,
      USER_NAME,
      USER_PASS);
  }
  return notificacioService;
}

```

Un cop inicialitzat el client ja es poden fer crides a les diferents operacions definides pel servei de notificació:

- RespostaAlta respostaAlta = getNotificacioService().alta(notificacio);
- RespostaConsultaEstatNotificacio respostaConsultaEstat =
getNotificacioService().consultaEstatNotificacio(identificador);

- RespostaConsultaEstatEnviament respostaConsultaEstat =
getNotificacioService().consultaEstatEnviament(referencia);

4.3. Exemple ús llibreria

```
package es.caib.app.plugin.caib.notificacio;

import java.util.ArrayList;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.List;
import javax.xml.datatype.DatatypeConfigurationException;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.datatype.XMLGregorianCalendar;
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.digest.DigestUtils;
import es.caib.notib.client.NotificacioRestClientFactory;
import es.caib.notib.ws.notificacio.Certificacio;
import es.caib.notib.ws.notificacio.Document;
import es.caib.notib.ws.notificacio.EntregaDeh;
import es.caib.notib.ws.notificacio.EntregaPostal;
import es.caib.notib.ws.notificacio.EntregaPostalTipusEnum;
import es.caib.notib.ws.notificacio.EntregaPostalViaTipusEnum;
import es.caib.notib.ws.notificacio.EnviamentTipusEnum;
import es.caib.notib.ws.notificacio.NotificacioService;
import es.caib.notib.ws.notificacio.PagadorCie;
import es.caib.notib.ws.notificacio.PagadorPostal;
import es.caib.notib.ws.notificacio.ParametresSeu;
import es.caib.notib.ws.notificacio.RespostaAlta;
import es.caib.notib.ws.notificacio.Notificacio;
import es.caib.notib.ws.notificacio.Enviament;
import es.caib.notib.ws.notificacio.RespostaConsultaEstatNotificacio;
import es.caib.notib.ws.notificacio.RespostaConsultaEstatEnviament;
import es.caib.app.plugin.SistemaExternException;
import es.caib.app.plugin.notificacio.EnviamentApp;
import es.caib.app.plugin.notificacio.EnviamentEstat;
import es.caib.app.plugin.notificacio.EnviamentReferenciaApp;
import es.caib.app.plugin.notificacio.NotificacioApp;
import es.caib.app.plugin.notificacio.NotificacioEstat;
import es.caib.app.plugin.notificacio.NotificacioPlugin;
```

```
import es.caib.app.plugin.notificacio.PersonaApp;
import es.caib.app.plugin.notificacio.RespostaConsultaEstatEnviamentApp;
import es.caib.app.plugin.notificacio.RespostaConsultaEstatNotificacioApp;
import es.caib.app.plugin.notificacio.RespostaEnviar;
import es.caib.app.plugin.utils.PropertiesHelper;

/**
 * Implementació de plugin d'enviament de notificacions emprant NOTIB.
 *
 * @author Limit Technologies <limit@limit.es>
 */
public class NotificacioPluginApp implements NotificacioPlugin {

    private NotificacioService notificacioService;

    @Override
    public RespostaEnviar enviar(NotificacioApp notificacio) throws SistemaExternException {
        try {
            Notificacio notificacioNotib = new Notificacio();
            notificacioNotib.setEmisorDir3Codi(notificacio.getEmisorDir3Codi());
            if (notificacio.getEnviamentTipus() != null) {
                switch (notificacio.getEnviamentTipus()) {
                    case COMUNICACIO:
                        notificacioNotib.setEnviamentTipus(EnviamentTipusEnum.COMUNICACIO);
                        break;
                    case NOTIFICACIO:
                        notificacioNotib.setEnviamentTipus(EnviamentTipusEnum.NOTIFICACIO);
                        break;
                }
            }
            notificacioNotib.setConcepte(notificacio.getConcepte());
            notificacioNotib.setDescripcio(notificacio.getDescripcio());
            notificacioNotib.setEnviamentDataProgramada(
                toXmlGregorianCalendar(notificacio.getEnviamentDataProgramada());
            notificacioNotib.setRetard(notificacio.getRetard());
            notificacioNotib.setCaducitat(toXmlGregorianCalendar(notificacio.getCaducitat()));
            Document document = new Document();
            document.setArxiuNom(notificacio.getDocumentArxiuNom());
            document.setContingutBase64(
                new String(Base64.encodeBase64(notificacio.getDocumentArxiuContingut()));
            document.setHash(
                new String(Base64.encodeBase64(
                    DigestUtils.sha256Hex(notificacio.getDocumentArxiuContingut()).getBytes())));
```

```
notificacioNotib.setDocument(document);
notificacioNotib.setProcedimentCodi(notificacio.getProcedimentCodi());
if (notificacio.getPagadorPostalDir3Codi() != null) {
    PagadorPostal pagadorPostal = new PagadorPostal();
    pagadorPostal.setDir3Codi(notificacio.getPagadorPostalDir3Codi());
    pagadorPostal.setContracteNum(notificacio.getPagadorPostalContracteNum());
    pagadorPostal.setContracteDataVigencia(
        toXmlGregorianCalendar(notificacio.getPagadorPostalContracteDataVigencia());
    pagadorPostal.setFacturacioClientCodi(notificacio.getPagadorPostalFacturacioClientCodi());
    notificacioNotib.setPagadorPostal(pagadorPostal);
}
if (notificacio.getPagadorCieDir3Codi() != null) {
    PagadorCie pagadorCie = new PagadorCie();
    pagadorCie.setDir3Codi(notificacio.getPagadorCieDir3Codi());
    pagadorCie.setContracteDataVigencia(
        toXmlGregorianCalendar(notificacio.getPagadorCieContracteDataVigencia());
    notificacioNotib.setPagadorCie(pagadorCie);
}
if (notificacio.getEnviaments() != null) {
    for (EnviamentApp enviament: notificacio.getEnviaments()) {
        Enviament enviamentNotib = new Enviament();
        enviamentNotib.setTitular(toPersonaNotib(enviament.getTitular()));
        if (enviament.getDestinatari() != null) {
            for (Persona destinatari: enviament.getDestinatari()) {
                enviamentNotib.getDestinatari().add(toPersonaNotib(destinatari));
            }
        }
        if (enviament.getEntregaPostalTipus() != null) {
            EntregaPostal entregaPostal = new EntregaPostal();
            switch (enviament.getEntregaPostalTipus()) {
                case NACIONAL:
                    entregaPostal.setTipus(EntregaPostalTipusEnum.NACIONAL);
                    break;
                case ESTRANGER:
                    entregaPostal.setTipus(EntregaPostalTipusEnum.ESTRANGER);
                    break;
                case APARTAT_CORREUS:
                    entregaPostal.setTipus(EntregaPostalTipusEnum.APARTAT_CORREUS);
                    break;
                case SENSE_NORMALITZAR:
                    entregaPostal.setTipus(EntregaPostalTipusEnum.SENSE_NORMALITZAR);
                    break;
            }
        }
    }
}
```

```
if (enviament.getEntregaPostalViaTipus() != null) {  
    switch (enviament.getEntregaPostalViaTipus()) {  
        case ALAMEDA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.ALAMEDA);  
            break;  
        case AVENIDA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.AVENIDA);  
            break;  
        case AVINGUDA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.AVINGUDA);  
            break;  
        case BARRIO:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.BARRIO);  
            break;  
        case BULEVAR:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.BULEVAR);  
            break;  
        case CALLE:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CALLE);  
            break;  
        case CALLEJA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CALLEJA);  
            break;  
        case CAMI:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CAMI);  
            break;  
        case CAMINO:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CAMINO);  
            break;  
        case CAMPO:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CAMPO);  
            break;  
        case CARRER:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CARRER);  
            break;  
        case CARRERA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CARRERA);  
            break;  
        case CARRETERA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CARRETERA);  
            break;  
        case CUESTA:  
            entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.CUESTA);
```



```
break;
case EDIFICIO:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.EDIFICIO);
    break;
case ENPARANTZA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.ENPARANTZA);
    break;
case ESTRADA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.ESTRADA);
    break;
case GLORIETA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.GLORIETA);
    break;
case JARDINES:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.JARDINES);
    break;
case JARDINS:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.JARDINS);
    break;
case KALEA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.KALEA);
    break;
case OTROS:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.OTROS);
    break;
case PARQUE:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PARQUE);
    break;
case PASAJE:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PASAJE);
    break;
case PASEO:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PASEO);
    break;
case PASSATGE:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PASSATGE);
    break;
case PASSEIG:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PASSEIG);
    break;
case PLACETA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PLACETA);
    break;
```

```
case PLAZA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PLAZA);
    break;
case PLAZUELA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PLAZUELA);
    break;
case PLAÇA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PLAÇA);
    break;
case POBLADO:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.POBLADO);
    break;
case POLIGONO:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.POLIGONO);
    break;
case PRAZA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.PRAZA);
    break;
case RAMBLA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.RAMBLA);
    break;
case RONDA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.RONDA);
    break;
case RUA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.RUA);
    break;
case SECTOR:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.SECTOR);
    break;
case TRAVESIA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.TRAVESIA);
    break;
case TRAVESSERA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.TRAVESSERA);
    break;
case URBANIZACION:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.URBANIZACION);
    break;
case VIA:
    entregaPostal.setViaTipus(EntregaPostalViaTipusEnum.VIA);
    break;
}
```

```

    }
    entregaPostal.setViaNom(enviament.getEntregaPostalViaNom());
    entregaPostal.setNumeroCasa(enviament.getEntregaPostalNumeroCasa());
    entregaPostal.setNumeroQualificador(
        enviament.getEntregaPostalNumeroQualificador());
    entregaPostal.setPuntKm(enviament.getEntregaPostalPuntKm());
    entregaPostal.setApartatCorreus(enviament.getEntregaPostalApartatCorreus());
    entregaPostal.setPortal(enviament.getEntregaPostalPortal());
    entregaPostal.setEscala(enviament.getEntregaPostalEscala());
    entregaPostal.setPlanta(enviament.getEntregaPostalPlanta());
    entregaPostal.setPorta(enviament.getEntregaPostalPorta());
    entregaPostal.setBloc(enviament.getEntregaPostalBloc());
    entregaPostal.setComplement(enviament.getEntregaPostalComplement());
    entregaPostal.setCodiPostal(enviament.getEntregaPostalCodiPostal());
    entregaPostal.setPoblacio(enviament.getEntregaPostalPoblacio());
    entregaPostal.setMunicipiCodi(enviament.getEntregaPostalMunicipiCodi());
    entregaPostal.setProvinciaCodi(enviament.getEntregaPostalProvinciaCodi());
    entregaPostal.setPaisCodi(enviament.getEntregaPostalPaisCodi());
    entregaPostal.setLinea1(enviament.getEntregaPostalLinea1());
    entregaPostal.setLinea2(enviament.getEntregaPostalLinea2());
    entregaPostal.setCie(enviament.getEntregaPostalCie());
    entregaPostal.setFormatSobre(enviament.getEntregaPostalFormatSobre());
    entregaPostal.setFormatFulla(enviament.getEntregaPostalFormatFulla());
    enviamentNotib.setEntregaPostal(entregaPostal);
}
if (enviament.getEntregaDehProcedimentCodi() != null) {
    EntregaDeh entregaDeh = new EntregaDeh();
    entregaDeh.setObligat(enviament.isEntregaDehObligat());
    entregaDeh.setProcedimentCodi(enviament.getEntregaDehProcedimentCodi());
    enviamentNotib.setEntregaDeh(entregaDeh);
}
notificacioNotib.getEnviaments().add(enviamentNotib);
}
}

ParametresSeu parametresSeu = new ParametresSeu();
parametresSeu.setExpedientSerieDocumental(notificacio.getSeuExpedientSerieDocumental());
parametresSeu.setProcedimentCodi(notificacio.getSeuProcedimentCodi());
parametresSeu.setExpedientUnitatOrganitzativa(
    notificacio.getSeuExpedientUnitatOrganitzativa());
parametresSeu.setExpedientIdentificadorEni(notificacio.getSeuExpedientIdentificadorEni());
parametresSeu.setExpedientSerieDocumental(notificacio.getSeuExpedientSerieDocumental());
parametresSeu.setExpedientTitol(notificacio.getSeuExpedientTitol());
parametresSeu.setRegistreOficina(notificacio.getSeuRegistreOficina());

```

```

parametresSeu.setRegistreLlibre(notificacio.getSeuRegistreLlibre());
parametresSeu.setRegistreOrgan(notificacio.getSeuRegistreOrgan());
parametresSeu.setIdioma(notificacio.getSeuIdioma());
parametresSeu.setAvisTitol(notificacio.getSeuAvisTitol());
parametresSeu.setAvisText(notificacio.getSeuAvisText());
parametresSeu.setAvisTextMobil(notificacio.getSeuAvisTextMobil());
parametresSeu.setOficiTitol(notificacio.getSeuOficiTitol());
parametresSeu.setOficiText(notificacio.getSeuOficiText());
notificacioNotib.setParametresSeu(parametresSeu);
  
```

```

RespostaAlta respostaAlta = getNotificacioService().alta(notificacioNotib);
  
```

```

if (respostaAlta.isError()) {
    throw new SistemaExternException(respostaAlta.getErrorDescripcio());
} else {
    RespostaEnviar resposta = new RespostaEnviar();
    if (respostaAlta.getEstat() != null) {
        switch (respostaAlta.getEstat()) {
            case PENDENT:
                resposta.setEstat(NotificacioEstat.PENDENT);
                break;
            case ENVIADA:
                resposta.setEstat(NotificacioEstat.ENVIADA);
                break;
            case FINALITZADA:
                resposta.setEstat(NotificacioEstat.FINALITZADA);
                break;
        }
    }
    resposta.setIdentificador(respostaAlta.getIdentificador());
    if (respostaAlta.getReferencies() != null) {
        List<EnviamentReferenciaApp> referencies = new ArrayList<EnviamentReferenciaApp>();
        for (EnviamentReferencia ref: respostaAlta.getReferencies()) {
            EnviamentReferenciaApp referencia = new EnviamentReferenciaApp();
            referencia.setTitularNif(ref.getTitularNif());
            referencia.setReferencia(ref.getReferencia());
            referencies.add(referencia);
        }
        resposta.setReferencies(referencies);
    }
    return resposta;
}
} catch (Exception ex) {
  
```

```

    throw new SistemaExternException(
        "No s'ha pogut enviar la notificació (" +
        "emisorDir3Codi=" + notificacio.getEmisorDir3Codi() + ", " +
        "enviamentTipus=" + notificacio.getEnviamentTipus() + ", " +
        "concepte=" + notificacio.getConcepte() + ")",
        ex);
  }
}

@Override
public RespostaConsultaEstatNotificacio consultarNotificacio(String identificador)
    throws SistemaExternException {
  try {
    RespostaConsultaEstatNotificacio respostaConsultaEstat =
      getNotificacioService().consultaEstatNotificacio(identificador);
    if (respostaConsultaEstat.isError()) {
      throw new SistemaExternException(respostaConsultaEstat.getErrorDescripcio());
    } else {
      RespostaConsultaEstatNotificacioApp resposta = new RespostaConsultaEstatNotificacioApp();
      if (respostaConsultaEstat.getEstat() != null) {
        switch (respostaConsultaEstat.getEstat()) {
          case PENDENT:
            resposta.setEstat(NotificacioEstat.PENDENT);
            break;
          case ENVIADA:
            resposta.setEstat(NotificacioEstat.ENVIADA);
            break;
          case FINALITZADA:
            resposta.setEstat(NotificacioEstat.FINALITZADA);
            break;
        }
      }
      return resposta;
    }
  } catch (Exception ex) {
    throw new SistemaExternException(
        "No s'ha pogut consultar l'estat de la notificació (" +
        "identificador=" + identificador + ")",
        ex);
  }
}

@Override

```

```
public RespostaConsultaEstatEnviament consultarEnviament(String referencia)
    throws SistemaExternException {

    try {

        RespostaConsultaEstatEnviament respostaConsultaEstat =
            getNotificacioService().consultaEstatEnviament(referencia);

        if (respostaConsultaEstat.isError()) {
            throw new SistemaExternException(respostaConsultaEstat.getErrorDescripcio());
        } else {

            RespostaConsultaEstatEnviamentApp resposta = new RespostaConsultaEstatEnviamentApp();

            if (respostaConsultaEstat.getEstat() != null) {
                switch (respostaConsultaEstat.getEstat()) {
                    case NOTIB_ENVIADA:
                        resposta.setEstat(EnviamentEstat.NOTIB_ENVIADA);
                        break;
                    case NOTIB_PENDENT:
                        resposta.setEstat(EnviamentEstat.NOTIB_PENDENT);
                        break;
                    case ABSENT:
                        resposta.setEstat(EnviamentEstat.ABSENT);
                        break;
                    case ADRESA_INCORRECTA:
                        resposta.setEstat(EnviamentEstat.ADRESA_INCORRECTA);
                        break;
                    case DESCONEGUT:
                        resposta.setEstat(EnviamentEstat.DESCONEGUT);
                        break;
                    case ENTREGADA_OP:
                        resposta.setEstat(EnviamentEstat.ENTREGADA_OP);
                        break;
                    case ENVIADA_CI:
                        resposta.setEstat(EnviamentEstat.ENVIADA_CI);
                        break;
                    case ENVIADA_DEH:
                        resposta.setEstat(EnviamentEstat.ENVIADA_DEH);
                        break;
                    case ENVIAMENT_PROGRAMAT:
                        resposta.setEstat(EnviamentEstat.ENVIAMENT_PROGRAMAT);
                        break;
                    case ERROR_ENTREGA:
                        resposta.setEstat(EnviamentEstat.ERROR_ENTREGA);
                        break;
                    case EXPIRADA:
                        resposta.setEstat(EnviamentEstat.EXPIRADA);
                        break;
                }
            }
        }
    }
}
```

```
case EXTRAVIADA:
    resposta.setEstat(EnviamentEstat.EXTRAVIADA);
    break;
case LLEGIDA:
    resposta.setEstat(EnviamentEstat.LLEGIDA);
    break;
case MORT:
    resposta.setEstat(EnviamentEstat.MORT);
    break;
case NOTIFICADA:
    resposta.setEstat(EnviamentEstat.NOTIFICADA);
    break;
case PENDENT_CIE:
    resposta.setEstat(EnviamentEstat.PENDENT_CIE);
    break;
case PENDENT_DEH:
    resposta.setEstat(EnviamentEstat.PENDENT_DEH);
    break;
case PENDENT_ENVIAMENT:
    resposta.setEstat(EnviamentEstat.PENDENT_ENVIAMENT);
    break;
case PENDENT_SEU:
    resposta.setEstat(EnviamentEstat.PENDENT_SEU);
    break;
case REBUTJADA:
    resposta.setEstat(EnviamentEstat.REBUTJADA);
    break;
case SENSE_INFORMACIO:
    resposta.setEstat(EnviamentEstat.SENSE_INFORMACIO);
    break;
}
}
resposta.setEstatData(toDate(respostaConsultaEstat.getEstatData()));
resposta.setEstatDescripcio(respostaConsultaEstat.getEstatDescripcio());
resposta.setEstatOrigen(respostaConsultaEstat.getEstatOrigen());
resposta.setReceptorNif(respostaConsultaEstat.getReceptorNif());
resposta.setReceptorNom(respostaConsultaEstat.getReceptorNom());
if (respostaConsultaEstat.getCertificacio() != null) {
    Certificacio certificacio = respostaConsultaEstat.getCertificacio();
    resposta.setCertificacioData(toDate(certificacio.getData()));
    resposta.setCertificacioOrigen(certificacio.getOrigen());
    resposta.setCertificacioContingut(
        Base64.decodeBase64(certificacio.getContingutBase64().getBytes()));
}
```

```

        resposta.setCertificacioHash(certificacio.getHash());
        resposta.setCertificacioMetadades(certificacio.getMetadades());
        resposta.setCertificacioCsv(certificacio.getCsv());
        resposta.setCertificacioTipusMime(certificacio.getTipusMime());
    }
    return resposta;
}
} catch (Exception ex) {
    throw new SistemaExternException(
        "No s'ha pogut consultar l'estat de l'enviament (" +
        "referencia=" + referencia + ")",
        ex);
}
}

private XMLGregorianCalendar toXmlGregorianCalendar(Date date)
    throws DatatypeConfigurationException {
    if (date == null) {
        return null;
    }
    GregorianCalendar gc = new GregorianCalendar();
    gc.setTime(date);
    return DatatypeFactory.newInstance().newXMLGregorianCalendar(gc);
}

private Date toDate(XMLGregorianCalendar calendar) throws DatatypeConfigurationException {
    if (calendar == null) {
        return null;
    }
    return calendar.toGregorianCalendar().getTime();
}

private Persona toPersonaNotib(
    PersonaApp persona) {
    Persona p = null;
    if (persona != null) {
        p = new Persona();
        p.setNif(persona.getNif());
        p.setNom(persona.getNom());
        p.setLlinatge1(persona.getLlinatge1());
        p.setLlinatge2(persona.getLlinatge2());
        p.setTelefon(persona.getTelefon());
        p.setEmail(persona.getEmail());
    }
    return p;
}
}

```



```
private NotificacioService getNotificacioService() {  
    if (notificacioService == null) {  
        notificacioService = NotificacioRestClientFactory.getRestClient(  
            getUrl(),  
            getUsername(),  
            getPassword());  
    }  
    return notificacioService;  
}  
  
private String getUrl() {  
    return PropertiesHelper.getProperties().getProperty(  
        "es.caib.app.plugin.notificacio.url");  
}  
  
private String getUsername() {  
    return PropertiesHelper.getProperties().getProperty(  
        "es.caib.app.plugin.notificacio.username");  
}  
  
private String getPassword() {  
    return PropertiesHelper.getProperties().getProperty(  
        "es.caib.app.plugin.notificacio.password");  
}  
}
```