



**Consorti
Administració Oberta
de Catalunya**

Plataforma iArxiu. Especificacions tècniques dels serveis web (WS). Mòdul d'ingrés



LOCALRET

Creat per: Consorci AOC
Versió: 4.5
Codi de referència: D1300 N-iArxiu/Manuals
Data: 2/2/2021

Índex

1	Introducció	3
2	Especificacions del servei ingrés	4
2.1	Operació <i>Ingest</i>	4
2.1.1	Paràmetres d'entrada	4
2.1.2	Paràmetres de sortida	6
2.2	Operació <i>ZipIngest</i>	7
2.2.1	Paràmetres d'entrada	7
2.2.2	Paràmetres de sortida	9
2.3	Operació <i>OfflineIngest</i>	9
2.3.1	Paràmetres d'entrada	9
2.3.2	Paràmetres de sortida	9
2.4	Operació <i>OfflineZipIngest</i>	9
2.4.1	Paràmetres d'entrada	9
2.4.2	Paràmetres de sortida	10
2.5	Operació <i>GetOfflineIngestStatus</i>	10
2.5.1	Paràmetres d'entrada	10
2.5.2	Paràmetres de sortida	10
2.6	Upload de fitxers grans	11
2.6.1	Operació <i>GetUploadTicket</i>	11
2.6.2	Upload servlet	11
2.6.3	Operació <i>OfflineUploadIngest</i>	12
2.7	Operació <i>GetOrganizationFondsAliasesByTA</i>	12
2.8	Tractament d'errors	14
3	Seguretat	15
4	Annexes	16
4.1	Taula d'il·lustracions	16
4.2	Fitxer WSDL	16

1 Introducció

El present document descriu els serveis web (en endavant WS) que s'han de seguir per al ingrés de paquets d'informació de transferència (PIT) a la plataforma iArxiu. La finalitat d'aquest document és detallar el WSDL del mòdul d'ingrés i descriure els diferents paràmetres que intervenen.

Les possibles alternatives per a realitzar un ingrés són les següents:

- ingrés amb fitxers incrustats en Base64 dins d'un METS
- ingrés d'un METS amb referències a fitxers binaris ubicats dins d'un ZIP.
- ingrés d'un METS amb referències a fitxers binaris enviats prèviament amb un servei de càrrega directa (no Web Service)

Per als dos primers casos es disposa de serveis d'ingrés online i serveis offline, però es recomana fer servir els serveis offline. A més a més, en funció del pes del PIT a transferir us recomanem les operacions del servei web:

Modalitats d'ingrés	Mida màxima permesa dels paquets d'informació de transferència (PIT)
Ingrés PIT amb documents en B64. Modalitat <i>online</i>	Fins a 5MB
Ingrés PIT amb referències als documents ubicats dins d'un fitxer ZIP. Modalitat <i>online</i>	Fins a 10 MB
Ingrés PIT amb documents en B64. Modalitat <i>offline</i>	Fins a 50 MB
Ingrés PIT amb referències a documents ubicats dins d'un fitxer ZIP. Modalitat <i>offline</i>	Fins a 200 MB
Ingrés PIT amb referències als documents enviats prèviament a un servei de càrrega directa	Fins a 300 MB

L'ingrés online torna al client directament l'identificador del paquet ingressat, mentre que l'ingrés offline torna un tiquet que posteriorment es pot fer servir per consultar l'estat de l'ingrés. L'avantatge de l'ingrés offline és que el client no ha d'esperar a que finalitzin tots els processos que desencadena el tractament d'un ingrés al core d'iArxiu.

El servei que ingressa binaris pujats amb la càrrega directa és adient per fitxers grans i sempre és en modalitat offline.

2 Especificacions del servei ingrés

Els WS es basen en el protocol SOAP i fan servir codificació de tipus *document/literal*.

Al punt 4.2 es detalla l'arxiu WSDL del mòdul d'ingrés. També es pot consultar a les següents urls:

Entorn	URL del servei
Preproducció	http://www.preproduccio.iarxiu.eocat.cat/core/soap/ingest.wsdl
Producció	http://www.iarxiu.eocat.cat/core/soap/ingest.wsdl

El WS d'ingrés consta de les següents operacions:

- Ingest
- ZipIngest
- OfflineIngest
- OfflineZipIngest
- GetOfflineIngestStatus
- GetUploadTicket
- OfflineUploadIngest
- GetOrganizationFondsAliasesByTA

A continuació s'enumeren els paràmetres d'entrada i sortida d'aquestes operacions:

2.1 Operació *Ingest*

Aquest mètode permet l'ingrés de PIT a la plataforma iArxiu de forma online incrustant els binaris en Base64 dins d'un fitxer METS.

2.1.1 Paràmetres d'entrada

Els paràmetres d'entrada (Request) d'aquesta operació són:

- mets: XML amb el paquet d'informació de transferència en format METS
- preservation: booleà per indicar si es vol fer ús del servei de preservació d'evidències
- contentTypeHandling: String per indicar quin tractament es desitja fer amb els ContentTypes (formats dels fitxers) dels fitxers
- signatureValidationDetails: booleà per indicar si es vol obtenir més informació sobre el resultat de la validació de les signatures electròniques que puguin estar presents en el METS.

L'esquema que descriu la *Request* és el següent:

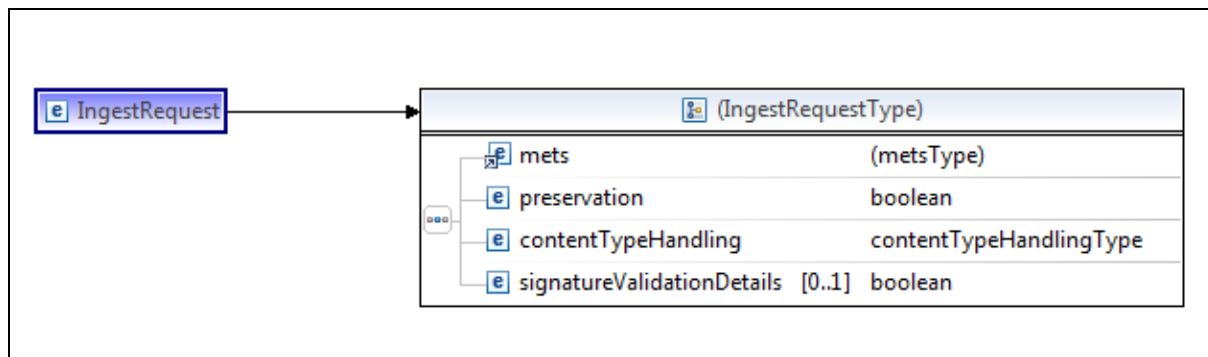


Figura 1: Esquema de la ingest request

A continuació es detallen els paràmetres:

2.1.1.1 mets

Aquest paràmetre conté el paquet en format XML segons es detalla al document: *Estructura i creació de Paquets d'Informació de Transferència (PIT) utilitzant el model METS¹*.

A més, i segons s'indica al fitxer WSDL, aquest xml seguirà l'esquema mets.xsd publicat a la web: <http://www.loc.gov/standards/mets/mets.xsd>

Els fitxers dels documents han d'estar incrustats en Base64 dins del METS.

2.1.1.2 preservation

Aquest paràmetre és un booleà que permet indicar si es vol fer ús del servei d'evidència en les electròniques presents en el METS.

Si el valor és igual a "true" és imprescindible que totes les signatures siguin vàlides en el moment de l'ingrés. En cas que en detecti alguna de no vàlida es refusa l'ingrés.

Si el valor és igual a "false" la plataforma acceptarà tant les signatures vàlides com les no vàlides.

Aquest valor és obligatori.

2.1.1.3 contentTypeHandling

El paràmetre contentTypeHandling permet especificar com iArxiu ha de tractar els valors de contentTypes informats pel client al METS. És obligatori que els clients emplenin l'atribut MIMETYPE dels fitxers presents al METS, però en funció del valor del paràmetre contentTypeHandling informat, la plataforma iArxiu realitza unes accions amb els valors informats pel client mitjançant les eines d'introspecció i detecció de formats que disposa (DROID i JHOVE).

Els possibles valors per al paràmetre contentTypeHandling són:

- replaceWithIntrospection: Es fa introspecció amb DROID/JHOVE de tots els fitxers. Si alguna d'aquestes eines identifica un format i és diferent al indicat pel client, llavors es substitueix pel detectat.

¹ Per a més informació sobre com construir els paquets d'informació de transferència en el format METS consulteu el següent document a: http://www.aoc.cat/index.php/ezwebin_site/Inici/SERVEIS/Gesti%C3%B3-interna/iArxiu/M%C3%A9s-informaci%C3%B3

- **completeWithIntrospection:** Es fa introspecció amb DROID/JHOVE de tots els fitxers però no es modifica el contentType inicial. Només s'informa al node de metadades object de PREMIS de cada fitxer amb les dades obtingudes amb DROID/JHOVE (quan hi ha coincidència). Quan no coincideix es respecte les dades que posa el client.
- **CheckAndReject:** Es fa introspecció amb DROID/JHOVE de tots els fitxers. Si DROID/JHOVE identifica un format i és diferent al indicat pel client, llavors es rebutja el paquet.

2.1.1.4 signatureValidationDetails

Aquest paràmetre és un booleà que permet indicar si es vol obtenir més informació sobre el resultat de la validació de les signatures electròniques que puguin estar presents en el METS.

2.1.2 Paràmetres de sortida

El paràmetre de sortida (Response) d'aquesta operació és un objecte *ingestInfo*:

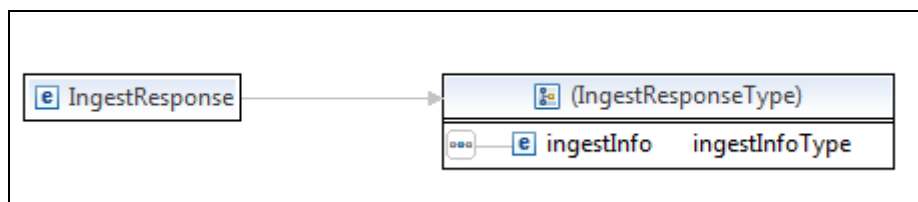


Figura 2: Esquema de la Ingest Response

L'objecte *ingestInfo* és del tipus *ingestInfoType* amb els següents elements:

- **id:** String que conté l'identificador de paquet en cas que l'ingrés hagi estat correcte.
- **preservedSignatures:** del tipus *preservedSignatureType*, String enumerat amb informació sobre la preservació de signatures. Els possibles valors són:
 - **none:** Indica que cap signatura s'ha pogut completar
 - **all:** Indica que totes les signatures s'han processat correctament
 - **onlyAdvanced:** Indica que s'han processat correctament les avançades i les simples no s'han pogut completar.
- **signatureValidationDetail:** Objecte del tipus *signatureValidationDetailType* que conté el detall de les signatures preservades, vàlides no preservades i invàlides.

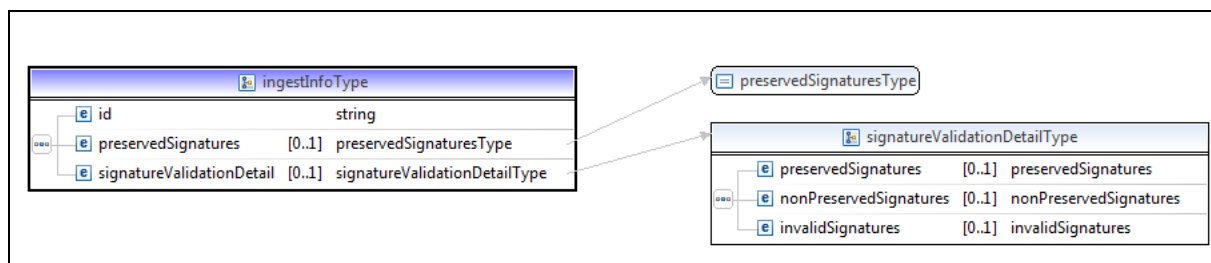


Figura 3: Esquema de la IngestResponseType

A continuació es mostra el detall de les signatures preservades (*preservedSignatures*), signatures vàlides no preservades (*nonPreservedSignatures*). La informació que retorna és l'identificador de la signatura



Figura 4: Esquema signatures preservades i signatures no preservades

I per últim, el detall de les signatures invàlides (*invalidSignatures*): llista de signatures del tipus *invalidSignatureType*. La informació que retorna és l'identificador de la signatura i el motiu de l'error de la validació.

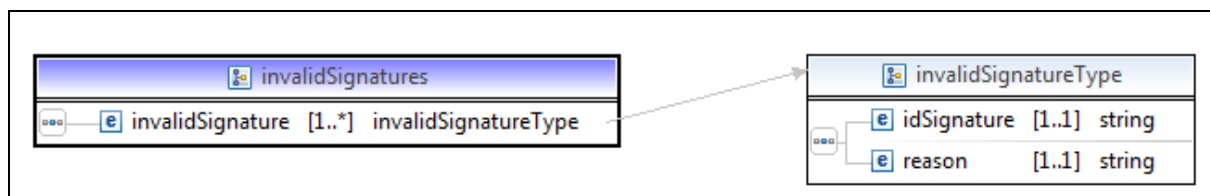


Figura 5: Esquema signatures invàlides

2.2 Operació *ZipIngest*

Aquest mètode permet l'ingrés de PIT a la plataforma iArxiu de forma online, de la mateixa manera que l'operació descrita al punt anterior, però els fitxers binaris es posen directament comprimits en un fitxer ZIP adjunt, en comptes d'estar incrustats en Base64 dins del METS.

2.2.1 Paràmetres d'entrada

Els paràmetres d'entrada (*Request*) d'aquesta operació són:

- *ZipFile*: Fitxer ZIP que conté un fitxer XML que representa el PIT en format METS i tots els fitxers a preservar
- *preservation*: booleà per indicar si es vol fer ús del servei d'evidència en les electròniques presents en el METS.
- *contentTypeHandling*: String per indicar quin tractament es desitja fer amb els ContentTypes (formats dels fitxers) dels fitxers.
- *signatureValidationDetails*: booleà per indicar si es vol obtenir més informació sobre el resultat de la validació de les signatures electròniques que puguin estar presents en el METS.

L'esquema que descriu la *Request* és el següent:

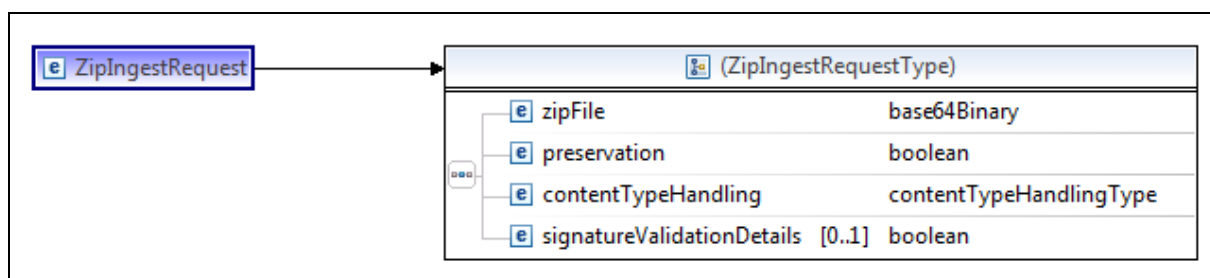


Figura 6: Esquema de la ZipIngest request

A continuació es detallen els paràmetres:

2.2.1.1 zipFile

Dins d'aquest paràmetre viatja un fitxer ZIP que conté:

- un fitxer anomenat mets.xml en format METS segons es detalla al document: Estructura i creació de Paquets d'Informació de Transferència (PIT) utilitzant el model METS²
- tots els fitxers que formen part del PIT

El fitxer mets.xml no ha de contenir els fitxers incrustats en Base64, sinó referències als fitxers que apareixen dins del ZIP.

A continuació es mostra una captura amb un exemple de fitxer ZIP:

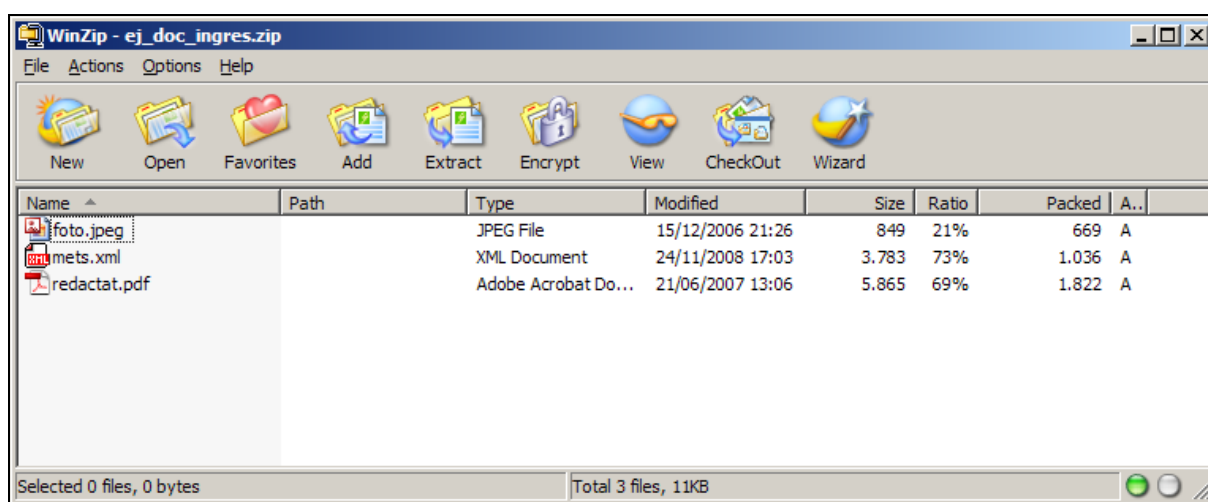


Figura 7: Contingut de fitxer ZIP

Seguint amb l'exemple, dins del fitxer mets.xml tindrem una referència a un fitxer extern com la següent:

```
<mets:FLocat LOCTYPE="URL" xlin:href="foto.jpeg" xmlns:xlin="http://www.w3.org/1999/xlink"/>
```

A l'exemple els fitxers es troben al mateix directori que el fitxer *mets.xml*, però també es poden posar en subdirectoris informant la ruta relativa corresponent al fitxer METS (per exemple: `xlin:href="imatges/foto.jpeg"`)

2.2.1.2 preservation

Aquest paràmetre té el mateix funcionament i significat que a la operació *Ingest*.

2.2.1.3 contentTypeHandling

Aquest paràmetre té el mateix funcionament i significat que a la operació *Ingest*.

2.2.1.4 signatureValidationDetails

Aquest paràmetre té el mateix funcionament i significat que a la operació *Ingest*.

² Per a més informació sobre com construir els paquets d'informació de transferència en el format METS consulteu el següent document a: http://www.aoc.cat/index.php/ezwebin_site/Inici/SERVEIS/Gesti%C3%B3-interna/iArxiu/M%C3%A9s-informaci%C3%B3

2.2.2 Paràmetres de sortida

Els paràmetres de sortida són els mateixos que a la operació *Ingest*.

2.3 Operació *OfflineIngest*

Aquest mètode permet l'ingrés de PIT a la plataforma iArxiu de forma similar a l'operació *Ingest*, però en mode offline. En aquesta modalitat la plataforma no processa l'ingrés de forma immediata, sinó que ho gestiona de forma indirecta, amb una gestió de cues.

En aquesta modalitat, la plataforma no retorna l'identificador de paquet sinó un tiquet que posteriorment s'ha de fer servir per obtenir l'estat de la petició.

2.3.1 Paràmetres d'entrada

Els paràmetres d'entrada són els mateixos que a la operació *Ingest*, exceptuant el paràmetre *signatureValidationDetails* que s'ha d'indicar en l'operació *GetOfflineIngestStatus*.

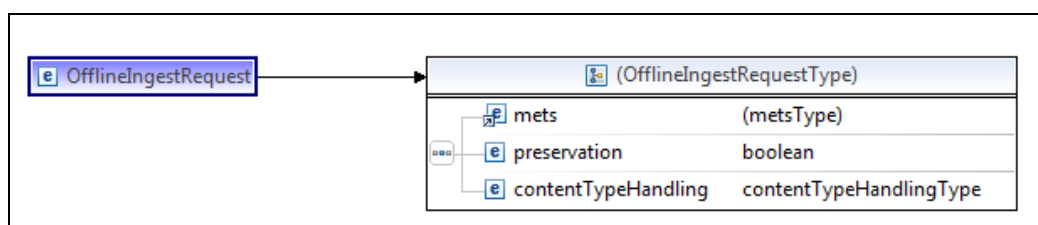


Figura 8: Esquema request offline Ingest

2.3.2 Paràmetres de sortida

Com a resposta (Response) l'operació torna un *String* amb un tiquet (identificador) que s'ha de fer servir amb la operació *GetOfflineIngestStatus* per consultar l'estat de l'ingrés.

2.4 Operació *OfflineZipIngest*

Aquest mètode permet l'ingrés de PIT a la plataforma iArxiu de forma similar a l'operació *ZipIngest*, però en mode offline. En aquesta modalitat la plataforma no processa l'ingrés de forma immediata, sinó que ho gestiona de forma indirecta, amb una gestió de cues.

En aquesta modalitat, la plataforma no retorna l'identificador de paquet sinó un tiquet que posteriorment s'ha de fer servir per obtenir l'estat de la petició.

2.4.1 Paràmetres d'entrada

Els paràmetres d'entrada són els mateixos que a la operació *ZipIngest*, exceptuant el paràmetre *signatureValidationDetails*. Aquest paràmetre s'incorpora dins de l'operació *GetOfflineIngestStatus*.

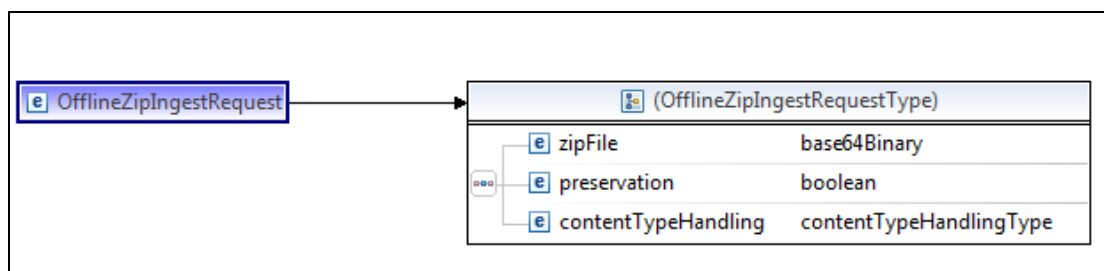


Figura 9: Esquema request offline Zip Ingest

2.4.2 Paràmetres de sortida

Com a resposta (Response) l'operació torna un *String* amb un tiquet (identificador) que s'ha de fer servir amb la operació *GetOfflineIngestStatus* per consultar l'estat de l'ingrés.

2.5 Operació *GetOfflineIngestStatus*

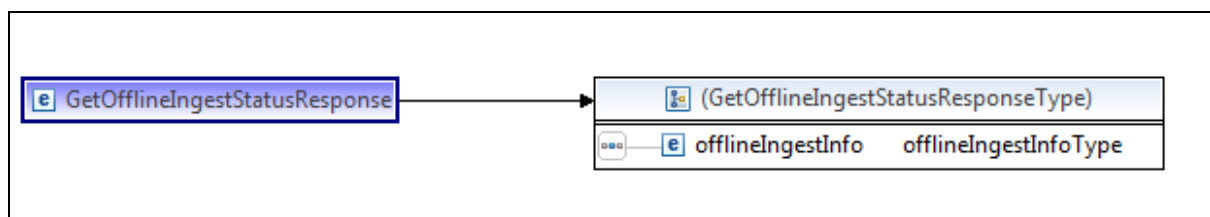
Aquesta operació permet consultar l'estat d'un ingrés a partir d'un tiquet dels retornats pels mètodes d'ingrés offline.

2.5.1 Paràmetres d'entrada

El paràmetres d'entrada és un *String* amb un tiquet i un atribut *signatureValidationdetails* indicant si es vol obtenir més informació sobre el resultat de la validació de les signatures electròniques que puguin estar presents en el METS.

2.5.2 Paràmetres de sortida

El paràmetre de sortida (Response) d'aquesta operació és del tipus *offlineIngestInfoType*:



Amb els següents paràmetres:

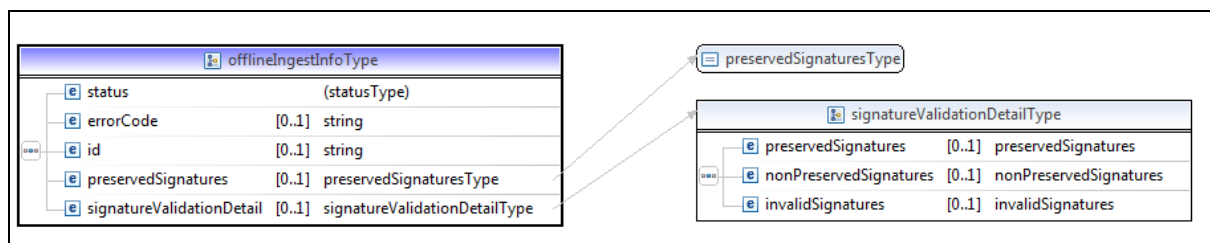


Figura 10: Esquema de la *GetOfflineIngestStatus Response*

- status: String enumerat amb l'estat actual de l'ingrés. Els possibles valors són:
 - ok
 - error
 - inProcess
 - okInvalidSignatures
- errorCode: String amb el codi d'error en cas d'error. La llista completa d'errors es troba al document: "Plataforma iArxiu. Especificacions tècniques del serveis web (WS). Tractament d'errors"
- id: String que conté l'identificador de paquet en cas de que l'ingrés hagi estat satisfactori
- preservedSignatures: String enumerat amb informació sobre la preservació de signatures. Els possibles valors són:

- none: Indica que cap signatura s'ha pogut completar
- all: Indica que totes les signatures s'han processat correctament
- onlyAdvanced: Indica que s'han processat correctament les avançades i les simples no s'han pogut completar.
- signatureValidationDetail: Objecte del tipus signatureValidationDetailType que conté el detall de les signatures preservades, vàlides no preservades i invàlides (veure punt 2.1.1.4).

2.6 Upload de fitxers grans

Les operacions que es detallen permeten realitzar una càrrega de fitxers grans sense haver de incrustar-los en una crida del WS. Es recomana fer servir aquesta opció per a fitxers de més de 10 Mb, encara que també es pot aplicar a fitxers més petits.

Per realitzar un ingrés mitjançant aquesta modalitat les passes són les següents:

- Mitjançant una crida a l'operació GetUploadTicket per obtenir un tiquet
- Invocar al servei d'upload amb els diferents binaris a ingressar (servlet)
- Realitzar una crida a l'operació OfflineUploadIngest
- Comprovar l'estat de l'ingrés amb GetOfflineIngestStatus

A continuació es detallen les operacions del WS:

2.6.1 Operació GetUploadTicket

Aquesta operació permet obtenir un tiquet per realitzar un ingrés mitjançant càrrega directa dels fitxers.

2.6.1.1 Paràmetres d'entrada

Aquesta operació no té cap paràmetre d'entrada.

2.6.1.2 Paràmetres de sortida

El paràmetre de sortida és un *String* amb un tiquet que posteriorment s'ha de fer servir per invocar el servei d'upload.

2.6.2 Upload servlet

El servlet d'upload permet fer l'enviament dels fitxers mitjançant l'estàndard HTTP. Els clients han d'implementar un formulari amb codificació "multipart/form-data" amb els següents paràmetres per fer l'enviament (mètode POST):

- tiquet: Identificador obtingut amb l'operació GetUploadTicket
- compressed: opcional, 'true' indica que farem un upload comprimit.
- tants elements "file" com fitxers es vulguin pujar.
 - Atenció: quan 'compressed' és 'true', llavors només podem enviar un 'file', que serà un Zip comprimit amb tots els fitxers, com el que espera el OfflineZipIngest

La URL del servlet en funció de l'entorn és:

Entorn	URL del servei
Preproducció	https://www.preproduccio.iarxiu.eacat.cat/core/servlet/upload
Producció	https://www.iarxiu.eacat.cat/core/servlet/upload

Si pugem els fitxers comprimits dins un zip (compressed='true')

El servlet només es pot cridar un cop, indicant-hi el fitxer zip comprimit amb tots els fitxers, com el que espera el `OfflineZipIngest`.

Si no fem un upload comprimit (no indiquem `compressed='true'`)

Aquest servlet es pot cridar més d'un cop amb un mateix tiquet, de forma que no cal enviar tots els fitxers en una mateixa crida.

Com que per un mateix tiquet tots els fitxers van a parar a la mateixa carpeta, s'han de tenir en compte les següents restriccions:

- el fitxer que representa el PIT s'ha de dir `mets.xml`
- els fitxers han de estar referenciats i al mateix directori que el `mets.xml`. És a dir, que no es poden fer servir directoris a l'atribut `xlin:href` del tag `Flocat` al `mets.xml`.

2.6.3 Operació `OfflineUploadIngest`

Aquest mètode desencadena l'ingrés dels fitxers prèviament pujats. Com que els fitxers poden ser grans, l'ingrés es fa amb la modalitat `offline`, de la mateixa manera que amb la operació `OfflineIngest`.

2.6.3.1 Paràmetres d'entrada

Els paràmetres d'entrada (`Request`) d'aquesta operació són:

- `uploadTicket`: Tiquet obtingut amb la operació `GetUploadTicket`.
- `preservation`: booleà per indicar si es vol fer ús del servei de preservació d'evidències.
- `contentTypeHandling`: `String` per indicar quin tractament es desitja fer amb els `ContentTypes` (formats dels fitxers) dels fitxers.

L'esquema que descriu la `Request` és:

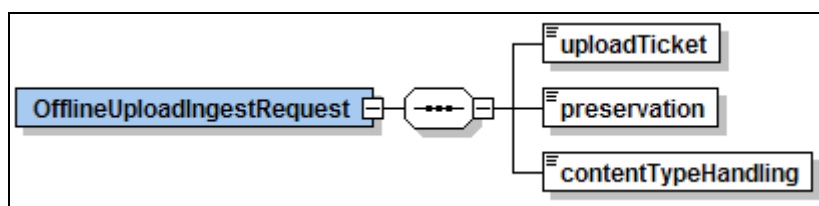


Figura 11: Esquema de la `OfflineUploadIngest` request

2.6.3.2 Paràmetres de sortida

De la mateixa manera que la operació `OfflineIngest`, aquesta operació torna un `String` amb un tiquet (identificador) que es pot de fer servir amb la operació `GetOfflineIngestStatus` per consultar l'estat de la petició.

2.7 Operació `GetOrganizationFondsAliasesByTA`

Aquesta operació permet obtenir informació dels ens i fons associats al TA corresponent al certificat amb el que es fa la crida.

S'obté la següent informació per a cada ens:

- Codi de l'ens
- Alies del ens
- Nom de l'ens
- Per a cada fons associat a l'ens:
 - Alies del fons

2.7.1.1 Paràmetres d'entrada

Aquesta operació no té cap paràmetre d'entrada.

2.7.1.2 Paràmetres de sortida

El paràmetre de sortida és un XML amb la següent estructura:

- **ing:organization:** tag per contenir la informació de cada ens
- **ing:code:** tag final amb el *Codi de l'ens*
- **ing:name:** tag final amb el *Nom de l'ens*
- **ing:cif:** tag final amb el *CIF de l'ens*
- **ing:fondsList:** tag per contenir la llista amb informació dels fons
- **ing:fonds:** tag per contenir la informació de cada fons
- **ing:alias:** tag final amb el *Alies de l'ens* o el *Alies del fons*

Per exemple, en el cas de que el TA estigués associat a dos ens, els alies dels quals són *aliesEns1* i *aliesEns2*, i on el primer ens té associat el fons amb alies *fons11* i el segon ens, dos fons amb alies *aliesFons21* i *aliesFons22*.

```
<ing:organization>
  <ing:code>codiEns1</ing:code>
  <ing:alias>aliesEns1</ing:alias>
  <ing:name>nomEns1</ing:name>
  <ing:cif>A00000000</ing:cif>
  <ing:fondsList>
    <ing:fonds>
      <ing:alias>aliesFons11</ing:alias>
    </ing:fonds>
  </ing:fondsList>
</ing:organization>
<ing:organization>
  <ing:code>codiEns2</ing:code>
  <ing:alias>aliesEns2</ing:alias>
  <ing:name>nomEns2</ing:name>
  <ing:cif>A00000000</ing:cif>
  <ing:fondsList>
    <ing:fonds>
      <ing:alias>aliesFons21</ing:alias>
    </ing:fonds>
    <ing:fonds>
      <ing:alias>aliesFons22</ing:alias>
    </ing:fonds>
  </ing:fondsList>
</ing:organization>.
```

2.8 Tractament d'errors

Si es produeix un error durant l'ingrés del PIT, la plataforma retorna una excepció que arriba al client com una *SOAP Fault* amb un codi d'error per identificar el tipus d'error que s'ha produït.

La llista completa d'errors es troba al document "*Plataforma iArxiu. Especificacions tècniques del serveis web (WS). Tractament d'errors*" consultable a:

http://www.aoc.cat/index.php/ezwebin_site/Inici/SERVEIS/Gesti%C3%B3-interna/iArxiu/Com-utilitzar-ho

3 Seguretat

La seguretat es basa en missatges SOAP signats digitalment per a la autenticació i assertions SAML per a la autorització, enviats per canals segurs (HTTPS).

Es poden consultar les especificacions detallades al document "*Plataforma iArxiu - Guia d'integració*" a: http://www.aoc.cat/index.php/ezwebin_site/Inici/SERVEIS/Gesti%C3%B3-interna/iArxiu/Com-utilitzar-ho

4 Annexes

4.1 Taula d'il·lustracions

Figura 1: Esquema de la ingest request	5
Figura 2: Esquema de la Ingest Response	6
Figura 3: Esquema de la IngestResponseType	6
Figura 4: Esquema signatures preservades i signatures no preservades	7
Figura 5: Esquema signatures invàlides	7
Figura 6: Esquema de la ZipIngest request	7
Figura 7: Contingut de fitxer ZIP	8
Figura 8: Esquema request offline Ingest	9
Figura 9: Esquema request offline Zip Ingest	9
Figura 10: Esquema de la GetOfflineIngestStatus Response	10
Figura 11: Esquema de la OfflineUploadIngest request	12

4.2 Fitxer WSDL

A continuació es mostra el contingut del fitxer WSDL que descriu el servei:


```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://schemas.core.iarxiu.hp.com/2.0/ingest" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:sch="http://schemas.core.iarxiu.hp.com/2.0/ingest"
  targetNamespace="http://schemas.core.iarxiu.hp.com/2.0/ingest">
  <wsdl:types>

    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:METS="http://www.loc.gov/METS/" xmlns="http://schemas.core.iarxiu.hp.com/2.0/ingest"
      targetNamespace="http://schemas.core.iarxiu.hp.com/2.0/ingest"
      elementFormDefault="qualified" attributeFormDefault="qualified">
      <xsd:import namespace="http://www.loc.gov/METS/"
        schemaLocation="http://www.loc.gov/standards/mets/mets.xsd" />
      <xsd:element name="IngestRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="METS:mets" />
            <xsd:element name="preservation" type="xsd:boolean" />
            <xsd:element name="contentTypeHandling" type="contentTypeHandlingType" />
            <xsd:element name="signatureValidationDetails" type="xsd:boolean"
              maxOccurs="1" minOccurs="0"/></xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="IngestResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="ingestInfo" type="ingestInfoType" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="ZipIngestRequest">
        <xsd:complexType>
```

```

        <xsd:sequence>
            <xsd:element name="zipFile" type="xsd:base64Binary" />
            <xsd:element name="preservation" type="xsd:boolean" />
            <xsd:element name="contentTypeHandling" type="contentTypeHandlingType" />
            <xsd:element name="signatureValidationDetails" type="xsd:boolean"
                maxOccurs="1" minOccurs="0"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="ZipIngestResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ingestInfo" type="ingestInfoType" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:simpleType name="contentTypeHandlingType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="replaceWithIntrospection" />
        <xsd:enumeration value="completeWithIntrospection" />
        <xsd:enumeration value="checkAndReject" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:element name="OfflineIngestRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="METS:mets" />
            <xsd:element name="preservation" type="xsd:boolean" />
            <xsd:element name="contentTypeHandling" type="contentTypeHandlingType" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="OfflineIngestResponse" type="xsd:string" />

```

```

<xsd:element name="OfflineZipIngestRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="zipFile" type="xsd:base64Binary" />
      <xsd:element name="preservation" type="xsd:boolean" />
      <xsd:element name="contentTypeHandling" type="contentTypeHandlingType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="OfflineZipIngestResponse" type="xsd:string" />
<xsd:element name="GetOfflineIngestStatusRequest">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="signatureValidationDetails"
          type="xsd:boolean" use="optional" default="false" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:element name="GetOfflineIngestStatusResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="offlineIngestInfo" type="offlineIngestInfoType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="OfflineUploadIngestRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="uploadTicket" type="xsd:string" />
      <xsd:element name="preservation" type="xsd:boolean" />
      <xsd:element name="contentTypeHandling" type="contentTypeHandlingType" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="OfflineUploadIngestResponse" type="xsd:string" />
<xsd:complexType name="offlineIngestInfoType">
    <xsd:sequence>
        <xsd:element name="status">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="ok" />
                    <xsd:enumeration value="error" />
                    <xsd:enumeration value="inProcess" />
                    <xsd:enumeration value="unknown" />
                    <xsd:enumeration value="okInvalidSignatures"></xsd:enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="errorCode" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="id" type="xsd:string" minOccurs="0" />
        <xsd:element name="preservedSignatures" type="preservedSignaturesType"
            minOccurs="0" />
        <xsd:element name="signatureValidationDetail" type="signatureValidationDetailType"
            maxOccurs="1" minOccurs="0"></xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ingestInfoType">
    <xsd:sequence>
        <xsd:element name="id" type="xsd:string" />
        <xsd:element name="preservedSignatures" type="preservedSignaturesType"
            minOccurs="0" />
        <xsd:element name="signatureValidationDetail" type="signatureValidationDetailType"
            maxOccurs="1" minOccurs="0"></xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="preservedSignaturesType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="all" />
            <xsd:enumeration value="onlyAdvanced" />
            <xsd:enumeration value="none" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="GetUploadTicketRequest">
        <xsd:complexType />
    </xsd:element>
    <xsd:element name="GetUploadTicketResponse">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="ticket" type="xsd:string">
                    <xsd:annotation>
                        <xsd:documentation>Ticket identifier for doing uploads
                    </xsd:documentation>
                    </xsd:annotation>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:complexType name="invalidSignatureType">
        <xsd:sequence>
            <xsd:element name="idSignature" type="xsd:string"
                maxOccurs="1" minOccurs="1">
            </xsd:element>
            <xsd:element name="reason" type="xsd:string" maxOccurs="1"
```

```
        minOccurs="1">
    </xsd:element>

</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="invalidSignature">
    <xsd:sequence>
        <xsd:element name="invalidSignature" type="invalidSignatureType">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="preservedSignature">
    <xsd:sequence>
        <xsd:element name="preservedSignature" type="xsd:string"
            maxOccurs="unbounded" minOccurs="1">
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="nonPreservedSignature">
    <xsd:sequence>
        <xsd:element name="nonPreservedSignature" type="xsd:string"></xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="signatureValidationDetailType">
    <xsd:sequence>
        <xsd:element name="preservedSignatures" type="preservedSignatures"
            maxOccurs="1" minOccurs="0">
```

```

        </xsd:element>
        <xsd:element name="nonPreservedSignatures" type="nonPreservedSignatures"
            maxOccurs="1" minOccurs="0">
        </xsd:element>
        <xsd:element name="invalidSignatures" type="invalidSignatures"
            maxOccurs="1" minOccurs="0"></xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="preservedSignatures">
    <xsd:sequence>
        <xsd:element name="preservedSignature" type="xsd:string"
            maxOccurs="unbounded" minOccurs="1"></xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="nonPreservedSignatures">
    <xsd:sequence>
        <xsd:element name="nonPreservedSignature" type="xsd:string"
            maxOccurs="unbounded" minOccurs="1"></xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="invalidSignatures">
    <xsd:sequence>
        <xsd:element name="invalidSignature" type="invalidSignatureType"
            maxOccurs="unbounded" minOccurs="1"></xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

</wsdl:types>
<wsdl:message name="ZipIngestResponse">

```



```

        <wsdl:part element="tns:ZipIngestResponse" name="ZipIngestResponse">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="GetUploadTicketResponse">
        <wsdl:part element="tns:GetUploadTicketResponse" name="GetUploadTicketResponse">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="IngestResponse">
        <wsdl:part element="tns:IngestResponse" name="IngestResponse">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="OfflineUploadIngestRequest">
        <wsdl:part element="tns:OfflineUploadIngestRequest" name="OfflineUploadIngestRequest">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="GetOfflineIngestStatusRequest">
        <wsdl:part element="tns:GetOfflineIngestStatusRequest" name="GetOfflineIngestStatusRequest">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="OfflineIngestResponse">
        <wsdl:part element="tns:OfflineIngestResponse" name="OfflineIngestResponse">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="GetOfflineIngestStatusResponse">
        <wsdl:part element="tns:GetOfflineIngestStatusResponse"
            name="GetOfflineIngestStatusResponse">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="OfflineIngestRequest">
        <wsdl:part element="tns:OfflineIngestRequest" name="OfflineIngestRequest">
        </wsdl:part>
    </wsdl:message>
    <wsdl:message name="OfflineZipIngestResponse">

```

```
<wsdl:part element="tns:OfflineZipIngestResponse" name="OfflineZipIngestResponse">
</wsdl:part>
</wsdl:message>
<wsdl:message name="OfflineZipIngestRequest">
  <wsdl:part element="tns:OfflineZipIngestRequest" name="OfflineZipIngestRequest">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="IngestRequest">
  <wsdl:part element="tns:IngestRequest" name="IngestRequest">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="OfflineUploadIngestResponse">
  <wsdl:part element="tns:OfflineUploadIngestResponse" name="OfflineUploadIngestResponse">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="GetUploadTicketRequest">
  <wsdl:part element="tns:GetUploadTicketRequest" name="GetUploadTicketRequest">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="ZipIngestRequest">
  <wsdl:part element="tns:ZipIngestRequest" name="ZipIngestRequest">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="Ingest">
  <wsdl:operation name="ZipIngest">
    <wsdl:input message="tns:ZipIngestRequest" name="ZipIngestRequest">
    </wsdl:input>
    <wsdl:output message="tns:ZipIngestResponse" name="ZipIngestResponse">
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="OfflineIngest">
    <wsdl:input message="tns:OfflineIngestRequest" name="OfflineIngestRequest">
    </wsdl:input>
```

```
        <wsdl:output message="tns:OfflineIngestResponse" name="OfflineIngestResponse">
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OfflineUploadIngest">
    <wsdl:input message="tns:OfflineUploadIngestRequest" name="OfflineUploadIngestRequest">
    </wsdl:input>
    <wsdl:output message="tns:OfflineUploadIngestResponse"
        name="OfflineUploadIngestResponse">
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="OfflineZipIngest">
    <wsdl:input message="tns:OfflineZipIngestRequest" name="OfflineZipIngestRequest">
    </wsdl:input>
    <wsdl:output message="tns:OfflineZipIngestResponse" name="OfflineZipIngestResponse">
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetOfflineIngestStatus">
    <wsdl:input message="tns:GetOfflineIngestStatusRequest"
        name="GetOfflineIngestStatusRequest">
    </wsdl:input>
    <wsdl:output message="tns:GetOfflineIngestStatusResponse"
        name="GetOfflineIngestStatusResponse">
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Ingest">
    <wsdl:input message="tns:IngestRequest" name="IngestRequest">
    </wsdl:input>
    <wsdl:output message="tns:IngestResponse" name="IngestResponse">
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUploadTicket">
    <wsdl:input message="tns:GetUploadTicketRequest" name="GetUploadTicketRequest">
    </wsdl:input>
```

```
        <wsdl:output message="tns:GetUploadTicketResponse" name="GetUploadTicketResponse">
        </wsdl:output>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="IngestSoap11" type="tns:Ingest">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="ZipIngest">
        <soap:operation soapAction="" />
        <wsdl:input name="ZipIngestRequest">
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="ZipIngestResponse">
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OfflineIngest">
        <soap:operation soapAction="" />
        <wsdl:input name="OfflineIngestRequest">
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="OfflineIngestResponse">
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="OfflineUploadIngest">
        <soap:operation soapAction="" />
        <wsdl:input name="OfflineUploadIngestRequest">
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output name="OfflineUploadIngestResponse">
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
</wsdl:service>
</wsdl:definitions>
```

```
</wsdl:operation>
<wsdl:operation name="OfflineZipIngest">
  <soap:operation soapAction="" />
  <wsdl:input name="OfflineZipIngestRequest">
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="OfflineZipIngestResponse">
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetOfflineIngestStatus">
  <soap:operation soapAction="" />
  <wsdl:input name="GetOfflineIngestStatusRequest">
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="GetOfflineIngestStatusResponse">
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="Ingest">
  <soap:operation soapAction="" />
  <wsdl:input name="IngestRequest">
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="IngestResponse">
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetUploadTicket">
  <soap:operation soapAction="" />
  <wsdl:input name="GetUploadTicketRequest">
    <soap:body use="literal" />
  </wsdl:input>
```

```
<wsdl:output name="GetUploadTicketResponse">
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="IngestService">
  <wsdl:port binding="tns:IngestSoap11" name="IngestSoap11">
    <soap:address
      location="https://www.preproduccio.iarxiu.eacat.cat/core/soap/ingest.wsdl" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```