

Manuale Titulus4Publication Web Service 4.4

- [Scopo](#)
- [Riferimenti](#)
- [URL del Servizio](#)
- [Autenticazione](#)
- [Metodi esposti](#)
- [Pubblicazione dei documenti dell'albo di ateneo](#)
- [Pubblicazione di generici documenti](#)
- [Consultazione dei documenti prodotti dagli export per le pubblicazioni](#)
- [Consultazione della consistenza di un fascicolo \(amministrazione trasparente\)](#)
- [Struttura delle risposte xml](#)
- [Esempi di xml restituiti](#)
 - [Pubblicazione del protocollo particolare \(repertorio "PP"\) del 2016](#)
 - [Pubblicazione dei documenti del 2016 che possiedono la voce di indice "Test voce di indice del 20/04/16"](#)
 - [Pubblicazione del 20/04/2016 dei documenti dell'albo di ateneo](#)
- [Version History](#)

Scopo

In questo documento si descrive il servizio *Titulus4Publication*, il cui scopo è quello di semplificare il recupero dei documenti da pubblicare sui portali degli atenei.

Il servizio Titulus4Publication viene distribuito nel pacchetto dei [web service di Titulus](#) a partire dalla versione **4.2.7**.

Riferimenti

Java DOC dei Web Service.

A questa documentazione si accede dalla [pagina di presentazione dei servizi](#).

URL del Servizio

L'URL del servizio Titulus4Publication ha la seguente struttura (<host> e <port> sono quelli della macchina che ospita i web service):

```
http://<host>:<port>/titulus_ws/services/Titulus4Publication
```

Il **WSDL** del servizio può essere recuperato al seguente URL (esso viene riportato anche nella [pagina di presentazione dei servizi](#)):

```
http://<host>:<port>/titulus_ws/services/Titulus4Publication?wsdl
```

Autenticazione

Si veda [Autenticazione](#).

Metodi esposti

- **alboDaPubblicare()**: consente di pubblicare i documenti del repertorio albo impostando eventualmente una data di pubblicazione.
- **documentiDaPubblicare()**: consente di pubblicare dei generici documenti impostando un intervallo - in anni - di pubblicazione e cercandoli o per voce di indice o per codice di repertorio.
- **documentiExportPubblicazione()**: consente la consultazione dei documenti prodotti dagli export per le pubblicazioni (es. amministrazione trasparente). Metodo presente a partire dalla versione **4.3.3** di Titulus.
- **consistenzaFascicoloTrasparenza()**: restituisce tutti i documenti di un fascicolo, compresi quelli contenuti nei suoi sottofascicoli (consistenza del fascicolo). Metodo presente a partire dalla versione **4.3.3** di Titulus.
- **currentTitlePage()/firstTitlePage()/lastTitlePage()/nextTitlePage()/prevTitlePage()/titlePage()**: consentono di scorrere le pagine dei result set prodotti dai metodi alboDaPubblicare(), documentiDaPubblicare() e consistenzaFascicoloTrasparenza(). Questi metodi sono simili a quelli esposti dal servizio Titulus4; per maggiori informazioni si veda "[Recupero documenti](#)".
- **getAttachment()**: consente il download dei file allegati ai documenti trovati. Questo metodo è identico a quello esposto dal servizio Titulus4; per maggiori informazioni si veda "[Download file associati ed immagini](#)".
- **getDocumentURL()**: consente di ottenere un URL di Titulus per accedere ad un documento dato un suo id. Questo metodo è identico a quello esposto dal servizio Titulus4; per maggiori informazioni si veda "[URL di un documento](#)".
- **getWsRelease()**: restituisce le informazioni sulla release del servizio.

Pubblicazione dei documenti dell'albo di ateneo

Il metodo alboDaPubblicare() consente di recuperare i documenti dell'albo ufficiale di ateneo che possono essere pubblicati sul portale dell'università.

Il metodo prende come parametro il bean AlboParams con cui è possibile:

- indicare una data di pubblicazione nei formati *aaaammgg* o *gg/mm/aaaa* [AlboParams → setDate(java.lang.String date)]; se omessa viene presa la data corrente;
- chiedere di avere solo la versione pdf dei file allegati ai documenti [AlboParams → setOnlyPdf(boolean onlyPdf)]. Questo parametro viene ignorato per i file firmati, dato che essi devono essere pubblicati nel formato originale (p7m, pdf e xml - sui portali i file firmati vanno pubblicati nel formato originale perché una versione pdf prodotta da Titulus perderebbe le informazioni di firma).

Di seguito si riporta un esempio in linguaggio Java, realizzato con la libreria [Axis](#). Lo stub client usato (nel codice “*titulus4Publication*”), è stato generato a partire dal wsdl del servizio, mediante il tool [WSDL2Java](#).

Nell'esempio viene effettuata l'autenticazione con l'utente “test-ws” (che deve essere registrato in ACL con login “test-ws”).

```
Titulus4PublicationServiceLocator t4psl = new Titulus4PublicationServiceLocator();

Titulus4Publication_PortType titulus4Publication = t4psl.getTitulus4Publication();

// impostazione della login per l'accesso autenticato al servizio
((Titulus4PublicationSoapBindingStub)titulus4Publication).setUsername("test-ws");
((Titulus4PublicationSoapBindingStub)titulus4Publication).setPassword("XXX");

...
AlboParams params = new AlboParams();

params.setDate("20/04/2016");
params.setOnlyPdf(true);

String firstPage = titulus4Publication.alboDaPubblicare(params);
...
```

Il metodo alboDaPubblicare() restituisce una risposta xml che rappresenta la prima pagina del result set dei documenti trovati (per maggiori informazioni si faccia riferimento a [Struttura delle risposte xml](#)). Con i metodi currentPage()/firstTitlePage()/lastTitlePage()/nextTitlePage()/prevTitlePage()/titlePage() è possibile scorrere le pagine del result set.

I documenti sono ordinati in modo decrescente secondo la data di protocollazione ed il numero di repertorio.

E', inoltre, possibile indicare anche il numero di documenti per pagina, portando così il default di 10 documenti al numero desiderato [AlboParams → setTitlePageSize(int titlePageSize)].

Pubblicazione di generici documenti

Il metodo documentiDaPubblicare() consente di recuperare dei generici documenti impostando un intervallo - in anni - di pubblicazione e cercandoli o per voce di indice o per codice di repertorio.

Il metodo prende come parametro il bean PublicationParams con cui è possibile:

- impostare un intervallo - in anni - di pubblicazione [PublicationParams → setStartYear(int startYear) e setEndYear(int endYear)] (questo intervallo fa riferimento all'anno di registrazione dei documenti);
- chiedere di avere solo la versione pdf dei file allegati ai documenti [AlboParams → setOnlyPdf(boolean onlyPdf)]. Questo parametro viene ignorato per i file firmati, dato che essi devono essere pubblicati nel formato originale (p7m, pdf e xml - sui portali i file firmati vanno pubblicati nel formato originale perché una versione pdf prodotta da Titulus perderebbe le informazioni di firma).
- cercare i documenti da pubblicare per voce di indice [PublicationParams → setIndexVoice(java.lang.String indexVoice)];
- cercare i documenti da pubblicare per codice di repertorio [PublicationParams → setRepCode(java.lang.String repCode)].

Occorre valorizzare almeno la voce di indice e/o il codice di repertorio per effettuare questa chiamata.

Di seguito si riporta un esempio in linguaggio Java, realizzato con la libreria [Axis](#). Lo stub client usato (nel codice “*titulus4Publication*”), è stato generato a partire dal wsdl del servizio, mediante il tool [WSDL2Java](#).

Nell'esempio viene effettuata l'autenticazione con l'utente “test-ws” (che deve essere registrato in ACL con login “test-ws”).

```

Titulus4PublicationServiceLocator t4psl = new Titulus4PublicationServiceLocator();

Titulus4Publication_PortType titulus4Publication = t4psl.getTitulus4Publication();

// impostazione della login per l'accesso autenticato al servizio
((Titulus4PublicationSoapBindingStub)titulus4Publication ).setUsername("test-ws");
((Titulus4PublicationSoapBindingStub)titulus4Publication ).setPassword("XXX");

...
PublicationParams params = new PublicationParams();

params.setOnlyPdf(true);
params.setRepCode("PP");
params.setEndYear(2016);
params.setStartYear(2015);

String firstPage = titulus4Publication.documentiDaPubblicare(params);
...

```

Il metodo `documentiDaPubblicare()` restituisce una risposta xml che rappresenta la prima pagina del result set dei documenti trovati (per maggiori informazioni si faccia riferimento a [Struttura delle risposte xml](#)). Con i metodi `currentTitlePage()/firstTitlePage()/lastTitlePage()/nextTitlePage()/prevTitlePage()/titlePage()` è possibile scorrere le pagine del result set.

I documenti sono ordinati in modo decrescente secondo la data di protocollazione, il numero di protocollo ed il numero di repertorio.

E', inoltre, possibile indicare anche il numero di documenti per pagina, portando così il default di 10 documenti al numero desiderato [`PublicationParams – > setTitlePageSize(int titlePageSize)`].

Consultazione dei documenti prodotti dagli export per le pubblicazioni

Il metodo `documentiExportPubblicazione()` consente di consultare i documenti prodotti dagli export per le pubblicazioni (es. per l'amministrazione trasparente).

Per individuare un export è necessario indicare un id di configurazione di export (`ExportPublicationParams.setConfigId(String)`).

Qualora si vogliano cercare dei documenti specifici si può indicare del testo libero per effettuare una ricerca full text sui campi dei documenti esportati (`ExportPublicationParams.setGlobalFullTextSearch(String)`). Oppure è possibile cercare per descrizione di repertorio (`ExportPublicationParams.setRepSearch(String)`), per data o intervallo di date (`ExportPublicationParams.setFromDateSearch(String)`, `ExportPublicationParams.setToDateSearch(String)`) o per id (`ExportPublicationParams.setPhysdocSearch(String)`).

Di default la chiamata restituisce la prima pagina del result set. Ogni pagina è composta da 10 documenti. E' possibile indicare un numero diverso di documenti per pagina (`ExportPublicationParams.setTitlePageSize(int)`). E' anche possibile richiedere una pagina specifica del result set (`ExportPublicationParams.setPageNumber(int)`). La prima pagina è la 1; il numero totale di pagine è contenuto nella risposta xml restituita.

Di ogni documento vengono restituiti i campi principali. Volendo è possibile richiedere solo i campi che interessano (`ExportPublicationParams.setFields(String[])`) o solo la versione pdf dei file, se presente (`ExportPublicationParams.setOnlyPdf(boolean)`). I documenti sono ordinati in modo decrescente secondo lo score dell'eventuale ricerca full text (score più alto in testa) , poi in modo decrescente in base a: codice ao, numero di repertorio, numero di protocollo e data di protocollazione.

Le pagine restituite dal metodo sono in formato XML. Volendo è possibile richiedere il formato JSON (`ExportPublicationParams.setJsonFormat(boolean)`).

Di seguito si riporta un esempio in linguaggio Java, realizzato con la libreria [Axis](#). Lo stub client usato (nel codice "`titulus4Publication`"), è stato generato a partire dal wsdl del servizio, mediante il tool [WSDL2Java](#).

Nell'esempio viene effettuata l'autenticazione con l'utente "test-ws" (che deve essere registrato in ACL con login "test-ws").

```

Titulus4PublicationServiceLocator t4psl = new Titulus4PublicationServiceLocator();

Titulus4Publication_PortType titulus4Publication = t4psl.getTitulus4Publication();

// impostazione della login per l'accesso autenticato al servizio
((Titulus4PublicationSoapBindingStub)titulus4Publication ).setUsername("test-ws");
((Titulus4PublicationSoapBindingStub)titulus4Publication ).setPassword("XXX");

...

ExportPublicationParams epp = new ExportPublicationParams();

// pagine di massimo 15 documenti
epp.setTitlePageSize(15);
epp.setConfigId("pubblicazione_0002");

String result = titulus4Publication.documentiExportPubblicazione(epp);

...

// richiesta della seconda pagina del result set
epp.setPageNumber(2);

// ricerca full text
epp.setGlobalFullTextSearch("documento allegati 2010-CINEBOL-0000057");

result = titulus4Publication.documentiExportPubblicazione(epp);

...

// ricerca full text
epp.setGlobalFullTextSearch("fattura tecnico computer");

// richiesta dei soli campi elencati
epp.setFields(new String[] { "/doc/oggetto", "/doc/oggetto/@*", "/doc/@nrecord", "/doc/rif_interni/rif/@*" });

result = titulus4Publication.documentiExportPubblicazione(epp);

...

// richiesta della risposta in json (invece del xml)
epp.setJsonFormat(true);

result = titulus4Publication.documentiExportPubblicazione(epp);

...

// ricerca per repertorio
epp.setRepSearch("albo ateneo");

result = titulus4Publication.documentiExportPubblicazione(epp);

...

// ricerca per range di date

epp.setFromDateSearch("20160101");
epp.setToDateSearch("20161231");

result = titulus4Publication.documentiExportPubblicazione(epp);

```



Per scorrere le pagine del result set occorre ripetere la chiamata al metodo `documentiExportPubblicazione()` impostando il numero di pagina richiesta nei parametri della chiamata (`ExportPublicationParams.setPageNumber(int)`).

La prima pagina è la 1; il numero totale di pagine è contenuto nella risposta xml restituita.

Consultazione della consistenza di un fascicolo (amministrazione trasparente)

Il metodo **consistenzaFascicoloTrasparenza** consente di ottenere tutti i documenti del fascicolo indicato, compresi quelli contenuti nei suoi sottofascicoli (consistenza del fascicolo).

E' necessario indicare il numero di un fascicolo o di un sottofascicolo. In caso di numero di un sottofascicolo viene ricavato il numero del fascicolo principale.

La chiamata può essere configurata (dal supporto [Titulus](#)) per restituire solo i *documenti principali* del fascicolo secondo i criteri stabiliti dall'ateneo.

Di seguito si riporta un esempio in linguaggio Java, realizzato con la libreria [Axis](#). Lo stub client usato (nel codice "*titulus4Publication*"), è stato generato a partire dal wsdl del servizio, mediante il tool [WSDL2Java](#).

Nell'esempio viene effettuata l'autenticazione con l'utente "test-ws" (che deve essere registrato in ACL con login "test-ws").

```
Titulus4PublicationServiceLocator t4psl = new Titulus4PublicationServiceLocator();

Titulus4Publication_PortType titulus4Publication = t4psl.getTitulus4Publication();

// impostazione della login per l'accesso autenticato al servizio
((Titulus4PublicationSoapBindingStub)titulus4Publication ).setUsername("test-ws");
((Titulus4PublicationSoapBindingStub)titulus4Publication ).setPassword("XXX");

...

String result = titulus4Publication.consistenzaFascicoloTrasparenza("2003-3DINBOL-04/03.00027", null);

...

// vengono richiesti solo i pdf (se presenti) dei documenti del fascicolo
BasicPublicationParams params = new BasicPublicationParams();

params.setOnlyPdf(true);

result = titulus4Publication.consistenzaFascicoloTrasparenza("2003-3DINBOL-04/03.00027.00001.00005", params);

// scorrimento delle pagine del result set
result = titulus4Publication.nextTitlePage();

...

result = titulus4Publication.nextTitlePage();

...
```

Struttura delle risposte xml

I metodi `alboDaPubblicare()`, `documentiDaPubblicare()`, `currentTitlePage()`, `firstTitlePage()`, `lastTitlePage()`, `nextTitlePage()`, `prevTitlePage()` e `titlePage()` restituiscono delle pagine che contengono [un elenco di documenti](#). Di ogni documento vengono riportati solo i dati che interessano la pubblicazione:

Campo	Percorsi XPath
id	/doc/@physdoc e /doc/@nrecord
codice AMMAOO	/doc/@cod_amm_aoo
numero di protocollo	/doc/@num_prot

data di protocollo	/doc/@data_prot
annullamento	/doc/@annullato
oggetto	/doc/oggetto
classificazione	/doc/classif e /doc/classif/@cod
codice di repertorio	/doc/repertorio/@cod
numero di repertorio	/doc/repertorio/@numero
descrizione di repertorio	/doc/repertorio
nome del RPA	/doc/rif_interni/rif[@diritto='RPA']/@nome_persona
ufficio del RPA	/doc/rif_interni/rif[@diritto='RPA']/@nome_uff
nome del mittente o dei destinatari	/doc/rif_esterni/rif/nome
codice dell'eventuale fascicolo	/doc/rif_interni/rif[@diritto='RPA']/@cod_fasc

Dei file di ogni documento vengono riportate solo le versioni più recenti e, in caso di file contenenti dati sensibili, solo le corrispondenti versioni con omissis (se presenti).

Su ogni file sono presenti i seguenti attributi:

- **name**: contiene l'id del file (da passare a *getAttachment()* per il download). *Esso è presente solo se il file può essere pubblicato.*
- **title**: contiene la descrizione del file. Questo attributo è presente anche quando manca l'attributo *name* perché il file non può essere pubblicato, in modo che tale descrizione possa essere sempre mostrata sul sito dell'università.
- **info**: contiene informazioni sul file e, quando esso non può essere pubblicato sul portale, contiene anche la ragione per cui non può essere scaricato.

Possibili valori dell'attributo *info*:

Valore	Significato
ANNULLATO	il documento è annullato quindi il file non è visibile (nessun file deve esserlo)

Se il file NON è firmato:

Valore	Significato
NORMAL	il file non contiene dati sensibili e può essere pubblicato
NORMAL,NO_PDF	il file non contiene dati sensibili ma è stato chiesto il pdf ed esso non è disponibile
NO_OMISSIS	il file contiene dati sensibili e non è disponibile una versione con omissis
OMISSIS,NO_PDF	il file rappresenta la versione con omissis di un file che contiene dati sensibili, ma è stato chiesto il pdf ed esso non è disponibile
OMISSIS	il file rappresenta la versione con omissis di un file che contiene dati sensibili

Se il file è firmato:

Valore	Significato
FIRMATO	il file è firmato (p7m, pdf o xml) e non contiene dati sensibili
FIRMATO,FORMATO_ERRATO	il file è firmato, non contiene dati sensibili ma il formato non è corretto (non è p7m, pdf o xml)
FIRMATO,NO_OMISSIS	il file è firmato, contiene dati sensibili e manca una versione con omissis
FIRMATO,OMISSIS	il file è firmato e rappresenta la versione con omissis di un file che contiene dati sensibili

FIRMATO,OMISSIS, FORMATO_ERRATO	il file è firmato e rappresenta la versione con omissis di un file che contiene dati sensibili, ma il formato non è corretto (non è p7m, pdf o xml)
--	---

Esempi di xml restituiti

Di seguito si riportano alcuni esempi di xml restituiti (tutte le chiamate sono state fatte impostando il parametro *setOnlyPdf* a *true*).

Pubblicazione del protocollo particolare (repertorio "PP") del 2016

Chiamata

```
PublicationParams params = new PublicationParams();

params.setOnlyPdf(true);
params.setRepCode("PP");
params.setEndYear(2016);
params.setStartYear(2016);

String firstPage = titulus4Publication.documentiDaPubblicare(params);
...
```

Risposta:

```
<Response xmlns:xw="http://www.kion.it/ns/xw" pageCount="1" pageIndex="1" seleId="5719e554ef343c2249fa53ec"
seleSize="1" query="(( ([doc_repertoriocod]=PP) AND ([doc_tipo]=arrivo)) OR ([doc_repertoriocod]=PP) AND
([doc_tipo]=partenza)) ) AND ([doc/@anno] = {2016|2016}) AND NOT([doc/@bozza]='&quot;si&quot;')">
  <doc physdoc="32987" nrecord="000033032-KIONBOL-a0eaa316-1b91-4lad-86e7-466eb63f436c" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000006" data_prot="20160420" annullato="no">
    <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 4</oggetto>
    <classif xml:space="preserve" cod="02/02">02/02 - Prorettori e delegati</classif>
    <repertorio cod="PP" numero="PP^KIONBOL-20160000001">Protocollo particolare in Partenza</repertorio>
    <rif_interni>
      <rif diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
    </rif_interni>
    <rif_esterni>
      <rif>
        <nome xml:space="preserve">Verdi Stefano</nome>
      </rif>
    </rif_esterni>
    <files>
      <xw:file name="Yh3AeWiMNORNhCgMHiYQNg==_000004562-FS_FILES-5c34aa7d-cb32-4a90-8c26-0d2120f2f787[1].xml"
title="IT00427620364_25.xml"
info="FIRMATO"/>
      <xw:file name="Mpzu3wCou/c0eWUw+adDKQ==_000004578-FS_FILES-659f9ccd-a832-475f-888d-0f4e908c9ee9[1].pdf"
title="Questa+è+una+prova+per+provare+il+modulo+FCA.pdf"
info="NORMAL"/>
    </files>
  </doc>
</Response>
```

Notare che, sebbene sia stata richiesta la versione pdf dei file, il primo file (che è scaricabile e quindi pubblicabile) è un xml perché è firmato.

Pubblicazione dei documenti del 2016 che possiedono la voce di indice "Test voce di indice del 20/04/16"

Chiamata

```
PublicationParams params = new PublicationParams();

params.setOnlyPdf(true);
params.setEndYear(2016);
params.setStartYear(2016);
params.setIndexVoice("Test voce di indice del 20/04/16");

String firstPage = titulus4Publication.documentiDaPubblicare(params);
...
```

Risposta:

```
<Response xmlns:xw="http://www.kion.it/ns/xw" pageCount="1" pageIndex="1" seleId="5719e555ef343c2249fa53f5"
seleSize="1" query="([/doc/voce_indice]=Test voce di indice del 20/04/16) AND ([/doc/@anno] = {2016|2016}) AND
NOT([/doc/@bozza]=&quot;si&quot;)">
  <doc physdoc="32984" nrecord="000033033-KIONBOL-94a637c4-c7b7-4db9-85d5-548cae941579" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000007" data_prot="20160420" annullato="no">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 5</oggetto>
  <classif xml:space="preserve" cod="04/02">04/02 - Atti di liberalità</classif>
  <rif_interni>
    <rif diritto="RPA" nome_persona="Bianchi Giorgio" nome_uff="Servizi Informatici"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">Alphabet S.p.A.</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file title="Manual_Utente_Titulus.pdf" info="NO_OMISSIS"/>
    <xw:file name="8xT9TJnxvzjaGgCAGz9jmQ==_000004585-FS_FILES-7165736b-fa6e-4d14-bbac-f8eb2e5a67aa[1].pdf"
      title="file_publicazioni_1_Publications.pdf"
      info="NORMAL"/>
  </files>
  <immagini>
    <xw:file title="Clouds, Birds, Moon, Venus.jpg" info="NORMAL,NO_PDF"/>
  </immagini>
</doc>
</Response>
```

Il file "Manual_Utente_Titulus.pdf" non è pubblicabile (quindi è privo dell'attributo *name*) perché contiene dati sensibili e manca una versione con omissis.

Il file "Clouds, Birds, Moon, Venus.jpg", invece, non è pubblicabile perché è un jpg e manca la versione in pdf.

Pubblicazione del 20/04/2016 dei documenti dell'albo di ateneo

Chiamata:

```
AlboParams params = new AlboParams();

params.setDate("20/04/2016");
params.setOnlyPdf(true);

String firstPage = titulus4Publication.alboDaPubblicare(params);
...
```

Risposta:

```
<Response xmlns:xw="http://www.kion.it/ns/xw" pageCount="1" pageIndex="1" seleId="5719e555ef343c2249fa53fe"
seleSize="7" query="([/doc/repertorio/@cod/]=&quot;ALBO&quot;) AND NOT([/doc/@bozza]=&quot;si&quot;) AND ([/doc
/publicazione/@dal/]={|20160420}) AND ([/doc/publicazione/@al/]={20160420|})">
  <doc physdoc="32992" nrecord="000033037-KIONBOL-e050687c-af0d-4676-878e-ae6c5e651fc2" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000010" data_prot="20160420" annullato="no">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 8</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-2016000007">Albo ufficiale di Ateneo arrivo</repertorio>
  <pubblicazione dal="20160420" al="20160505"/>
  <rif_interni>
    <rif diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">ACME agenzia giornalistica italia</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file name="PssD4spIzRwHilKtCJVLXg==_000004582-FS_FILES-a5b9c1df-c519-4efc-9669-448c20274d26[1].pdf"
      title="tif_monopagina.pdf"
      info="NORMAL"/>
  </files>
```



```

</files>
</doc>
<doc physdoc="32985" nrecord="000033036-KIONBOL-af618923-24d4-4756-9878-2c6ac63f0ee2" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000009" data_prot="20160420" annullato="no">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 7</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-20160000006">Albo ufficiale di Ateneo arrivo</repertorio>
  <pubblicazione dal="20160420" al="20160505"/>
  <rif_interni>
    <rif diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">ACME agenzia giornalistica italia</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file title="Il piatto di oggi.docx" info="FIRMATO,FORMATO_ERRATO"/>
    <xw:file title="Test del Marchese del Grillo.docx" info="OMISSIS,FIRMATO,FORMATO_ERRATO"/></files>
</doc>
<doc physdoc="32993" nrecord="000033035-KIONBOL-089a4db6-b135-48d8-80f3-9763c461024d" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000008" data_prot="20160420" annullato="si">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 6</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-20160000005">Albo ufficiale di Ateneo arrivo</repertorio>
  <pubblicazione dal="20160420" al="20160505"/>
  <rif_interni>
    <rif diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">Apple S.p.A.</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file title="doc test.[49152].D del Centro Int ELFID pdf.p7m" info="ANNULLATO"/>
    <xw:file title="Documento di progetto.docx" info="ANNULLATO"/>
  </files>
</doc>
<doc physdoc="32989" nrecord="000033031-KIONBOL-41d824c2-1850-443d-9811-a21af2affa7f" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000005" data_prot="20160420" annullato="no">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 3</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-20160000004">Albo ufficiale di Ateneo arrivo</repertorio>
  <pubblicazione dal="20160420" al="20160505"/>
  <rif_interni>
    <rif diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">Alphabet S.p.A.</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file name="a5C/qC7ejcBHiRJeq/HXWQ==_000004568-FS_FILES-7627bfa1-d35e-4e5f-8128-dae0a4095a9f[1].docx.p7m"
      title="File_firmato_doc.docx.p7m"
      info="FIRMATO"/>
    <xw:file title="prova_firmato.pdf" info="FIRMATO,NO_OMISSIS"/>
    <xw:file name="mVnQ+VGzYdGhGB2f/+dSbQ==_000004569-FS_FILES-653a06b1-f4dc-4754-b47e-94a6b6a7b391[2].pdf"
      title="prova_firmato 2.pdf"
      info="FIRMATO"/>
    <xw:file name="tRhSXzJR4g39dVO960Bu5w==_000004570-FS_FILES-b117e164-4915-4598-8d18-03e714f5f921[1].pdf.p7m"
      title="IT00427620364_25 OMISSIS"
      info="OMISSIS,FIRMATO"/></files>
</doc>
<doc physdoc="32988" nrecord="000033030-KIONBOL-c9cdc86c-f6a2-4bd6-a0b4-1b6cdf4fe9c9" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000004" data_prot="20160420" annullato="no">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 2</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-20160000003">Albo ufficiale di Ateneo arrivo</repertorio>

```

```
<pubblicazione dal="20160420" al="20560505"/>
<rif_interni>
  <rif_diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
</rif_interni>
<rif_esterni>
  <rif>
    <nome xml:space="preserve">ACME agenzia giornalistica italia</nome>
  </rif>
</rif_esterni>
<files>
  <xw:file name="hV0+Qd7fQeccmxdquhp0uQ==_000004548-FS_FILES-8c3cb601-a976-489e-9a7a-053038alb472[9].pdf"
    title="Manuale_ACL.pdf"
    info="NORMAL"/>
  <xw:file name="7L2qSEHfWdJijbom02cn4g==_000004587-FS_FILES-52b60b2a-51df-41bc-9736-540fe153f234[1].pdf"
    title="Questa è una prova per provare il modulo FCA OMISSIS.pdf"
    info="OMISSIS"/>
  <xw:file name="qzjI58V2HX+FMK2qiGzrCg==_000004588-FS_FILES-90b8df5d-aa0f-404b-9345-dc9486cc735c[1].pdf"
    title="tif_monopagina.pdf"
    info="NORMAL"/>
  <xw:file name="Zif1jnnPqBq58NDIRftwjA==_000004557-FS_FILES-ec85d041-6980-4cb0-8966-22027c48a710[2].pdf"
    title="cool.pdf"
    info="OMISSIS"/></files>
</doc>
<doc physdoc="32991" nrecord="000033029-KIONBOL-aaa560b9-da77-40bb-8768-05339622ea31" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000003" data_prot="20160420" annullato="no">
  <oggetto xml:space="preserve">Test per il servizio Titulus4Publication numero 1</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-20160000002">Albo ufficiale di Ateneo arrivo</repertorio>
  <pubblicazione dal="20160420" al="20560505"/>
  <rif_interni>
    <rif_diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">Apple S.p.A.</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file title="Test manutenzione.docx" info="NORMAL,NO_PDF"/>
    <xw:file title="all 4 - Ordinanza Giudice Lavoro 14-1-2015.zip" info="NO_OMISSIS"/>
    <xw:file name="D4rlqj79+htU7HYRWbOSDPg==_000004589-FS_FILES-2edcf252-1004-44ab-a094-b3a793f21c0b[1].pdf"
      title="tif_monopagina_ruotato.pdf"
      info="NORMAL"/>
    <xw:file title="prova-utf8 con omissis" info="OMISSIS,NO_PDF"/></files>
</doc>
<doc physdoc="32990" nrecord="000033028-KIONBOL-1c5ef45a-6e00-4b10-8f47-49fb00fe2d45" anno="2016"
cod_amm_aoo="KIONBOL" num_prot="2016-KIONBOL-0000002" data_prot="20160413" annullato="no">
  <oggetto xml:space="preserve">Test albo di ateneo numero 1 del 13/04/16</oggetto>
  <classif xml:space="preserve" cod="01/07">01/07 - Archivio</classif>
  <repertorio cod="ALBO" numero="ALBO^KIONBOL-20160000001">Albo ufficiale di Ateneo arrivo</repertorio>
  <pubblicazione dal="20160413" al="20570428"/>
  <rif_interni>
    <rif_diritto="RPA" nome_persona="Rossi Mario" nome_uff="Servizio Tecnico"/>
  </rif_interni>
  <rif_esterni>
    <rif>
      <nome xml:space="preserve">Facebook Inc.</nome>
    </rif>
  </rif_esterni>
  <files>
    <xw:file name="Xet+/XqLhLHbRVu2gndCbg==_000004590-FS_FILES-286a872d-8e2a-4298-aa34-ad8d86203612[1].pdf"
      title="prova-utf8.pdf"
      info="NORMAL"/>
    <xw:file name="oJ9t1I3YdqdbxjS+GNV2wg==_000004592-FS_FILES-d125bf9d-aff8-4f9d-9b85-6bd623c3a0f8[1].pdf"
      info="OMISSIS"
      title="Madagascar.odt"/>
  </files>
</doc>
</Response>
```

Il secondo documento (con @physdoc = "32985") contiene due file firmati in un formato non riconosciuto (cioè diverso da p7m, pdf o xml), quindi questi file non sono pubblicabili (in realtà è difficile che si verifichi un caso del genere perché Titulus non considera come firmati i file che hanno un formato diverso da p7m, pdf e xml).

Invece il terzo documento (con @physdoc = "32993"), essendo annullato, ha i file che non sono pubblicabili.

Version History

[Version history dei servizi web di Titulus 4.3](#)