

Manuale Accesso al servizio TitulusOrgani 4.5 in modalità RESTful

- [Accesso al servizio TitulusOrgani 4.5 in modalità RESTful](#)
- [Autenticazione](#)
- [Diritto di consultazione delle sedute](#)
- [Tipologia delle risposte restituite dal servizio](#)
 - [Errori legati all'utente usato per l'autenticazione](#)
- [Funzionalità esposte](#)
 - [Consultazione degli organi di un ateneo](#)
 - [URL](#)
 - [Metodo di richiesta supportato](#)
 - [Parametri della chiamata](#)
 - [Risposta della chiamata](#)
 - [Elenco delle sedute di un organo](#)
 - [URL](#)
 - [Metodo di richiesta supportato](#)
 - [Parametri della chiamata](#)
 - [Risposta della chiamata](#)
 - [Consultazione dell'odg di una seduta](#)
 - [URL](#)
 - [Metodo di richiesta supportato](#)
 - [Parametri della chiamata](#)
 - [Risposta della chiamata](#)
 - [Download di un file](#)
 - [URL](#)
 - [Metodo di richiesta supportato](#)
 - [Parametri della chiamata](#)
 - [Risposta della chiamata](#)

Accesso al servizio TitulusOrgani 4.5 in modalità RESTful

Il servizio TitulusOrgani4 espone alcune funzionalità di consultazione di organi e sedute accessibili in modalità RESTful.

In questa pagina si riportano le chiamate che è possibile effettuare.

Autenticazione

Tutte le chiamate al servizio devono essere autenticate. Infatti lo username indicato in fase di autenticazione viene utilizzato dal servizio per l'accesso all'archivio di organi secondo un profilo utente che deve essere registrato nell'anagrafica di Titulus (ACL). Allo username indicato deve corrispondere *un solo profilo utente*.

I web service di Titulus sono indipendenti dall'autenticazione usata. E' quindi possibile impiegare la Basic Authentication di Tomcat, Shibboleth o altro.

Diritto di consultazione delle sedute

Affinché sia possibile consultare gli organi e le loro sedute, è necessario che nel profilo dell'utente usato dal client sia attivato l'opportuno diritto di pubblicazione come indicato in [Consultazione delle sedute](#).

Tipologia delle risposte restituite dal servizio

Ogni chiamata al servizio restituisce al client una risposta in formato XML. Il dettaglio di queste risposte viene riportato nei paragrafi successivi, in cui si illustrano le singole funzionalità esposte.

Anche in caso di errore vengono inviate al client delle risposte in formato XML, associate ai codici di errore http 400 e 500.

Una generica risposta di errore ha la seguente struttura:

```
<Response result="error">
  <errore cod="...">
    <descrizione>...</descrizione>
    [ <eccezione>STACK_TRACE</eccezione> ]
    [additionalFields]
  </errore>
</Response>
```

Ogni errore è identificato da un codice univoco (*/Response/errore/@cod*) e da una descrizione (*/Response/errore/descrizione*).

In caso di errori non previsti viene indicato lo stack trace dell'errore che si è verificato; a seconda della tipologia di errore, inoltre, è possibile avere dei campi addizionali (*additionalFields*) che forniscono ulteriori dettagli sull'errore.

L'encoding usato per tutte le risposte XML è UTF-8.

Errori legati all'utente usato per l'autenticazione

Di seguito vengono riportate le 3 tipologie di errore legate all'utente usato per l'autenticazione delle richieste http.

Caso di utente remoto non autenticato (la richiesta http non è stata autenticata):

```
<Response result="error">
  <errore cod="WSTOR_E001">
    <descrizione>Eccezione non prevista</descrizione>
    <eccezione>java.lang.Exception: Utente remoto non autenticato
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:128)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:93)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:63)
      at it.kion.titulus.ws.TitulusOrgani4.elencoOrgani(TitulusOrgani4.java:320)
      at it.kion.titulus.ws.restful.organi.Organi.doGet(Organi.java:115)
      at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
      at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
      at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
      at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
      at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
      at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
      at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:470)
      at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
      at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
      at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
      at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
      at org.apache.coyote.http11.Http11AprProcessor.process(Http11AprProcessor.java:861)
      at org.apache.coyote.http11.Http11AprProtocol$Http11ConnectionHandler.process(Http11AprProtocol.java:
579)
      at org.apache.tomcat.util.net.AprEndpoint$Worker.run(AprEndpoint.java:1584)
      at java.lang.Thread.run(Thread.java:662)
    </eccezione>
  </errore>
</Response>
```

Caso di utente non trovato nell'anagrafica di Titulus:

```

<Response result="error">
  <errore cod="WSTOR_E001">
    <descrizione>Eccezione non prevista</descrizione>
    <eccezione>java.lang.Exception: User not found: user 'test-ws' does not exist!
      at it.kion.titulus.ws.Titulus4.init(Titulus4.java:355)
      at it.kion.titulus.ws.BasicMinorService.init(BasicMinorService.java:76)
      at it.kion.titulus.ws.TitulusOrgani4.init(TitulusOrgani4.java:178)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:141)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:93)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:63)
      at it.kion.titulus.ws.TitulusOrgani4.elencoOrgani(TitulusOrgani4.java:320)
      at it.kion.titulus.ws.restful.organi.Organi.doGet(Organi.java:115)
      at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
      at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
      at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
      at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
      at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
      at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
      at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:563)
      at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
      at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
      at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
      at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
      at org.apache.coyote.http11.Http11AprProcessor.process(Http11AprProcessor.java:861)
      at org.apache.coyote.http11.Http11AprProtocol$Http11ConnectionHandler.process(Http11AprProtocol.java:
579)
      at org.apache.tomcat.util.net.AprEndpoint$Worker.run(AprEndpoint.java:1584)
      at java.lang.Thread.run(Thread.java:662)
    Caused by: java.sql.SQLException: Utente non riconosciuto
      at it.highwaytech.apps.code.PersonalRights.extractAllPersonalRights(PersonalRights.java:471)
      at it.highwaytech.apps.code.PersonalRights.extractAllPersonalRights(PersonalRights.java:442)
      at it.kion.titulus.ws.PRights.<init>(PRights.java:271)
      at it.kion.titulus.ws.Titulus4.init(Titulus4.java:309)
      ... 22 more
    </eccezione>
  </errore>
</Response>

```

Caso di più profili associati alla login indicata (viene accettato un solo profilo):

```

<Response result="error">
  <errore cod="WSTOR_E001">
    <descrizione>Eccezione non prevista</descrizione>
    <eccezione>java.lang.Exception: La login 'test-ws' individua più profili, mentre è richiesto che ad essa ne
corrisponda uno solo!
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:148)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:93)
      at it.kion.titulus.ws.TitulusService.checkRemoteUser(TitulusService.java:63)
      at it.kion.titulus.ws.TitulusOrgani4.elencoOrgani(TitulusOrgani4.java:320)
      at it.kion.titulus.ws.restful.organi.Organi.doGet(Organi.java:115)
      at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
      at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
      at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
      at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
      at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
      at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
      at org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:563)
      at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
      at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
      at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
      at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
      at org.apache.coyote.http11.Http11AprProcessor.process(Http11AprProcessor.java:861)
      at org.apache.coyote.http11.Http11AprProtocol$Http11ConnectionHandler.process(Http11AprProtocol.java:
579)
      at org.apache.tomcat.util.net.AprEndpoint$Worker.run(AprEndpoint.java:1584)
      at java.lang.Thread.run(Thread.java:662)
    </eccezione>
  </errore>
</Response>

```

Funzionalità esposte

Consultazione degli organi di un ateneo

URL

http://host/titulus_ws/organi/

Metodo di richiesta supportato

GET

Parametri della chiamata

- **aoo** (obbligatorio): rappresenta il codice AMMAOO degli organi che si vogliono consultare.

Esempio di invocazione:

http://host/titulus_ws/organi/?aoo=UNKNCLE

Risposta della chiamata

Di seguito si riporta un esempio di risposta XML in cui sono presenti 2 organi: il Consiglio di Amministrazione ed il Senato Accademico.

```

<Response size="2">
  <organo cod="CLE-CdA" cod_amm_aoo="UNKNCLE">
    <nome xml:space="preserve">Consiglio di Amministrazione</nome>
    <componenti>
      <componente>
        <nominativo xml:space="preserve">Prof. Mario Rossi</nominativo>
        <incarico xml:space="preserve">Presidente</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Cesare Verdi</nominativo>
        <incarico xml:space="preserve">Vicepresidente</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Stefano Stefani</nominativo>
        <incarico xml:space="preserve">Rappresentante Rettore dell'Università degli Studi di Padova</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Leonardo Marroni</nominativo>
        <incarico xml:space="preserve">Rappresentante Rettore dell'Università di Firenze</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Giuseppe Bianchi</nominativo>
        <incarico xml:space="preserve">Rappresentante Rettore dell'Università degli Studi di Napoli, Federico
II</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Armando Turchesi</nominativo>
        <incarico xml:space="preserve">Rappresentante del Rettore dell'Università di Venezia</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Dott. Marco Viola</nominativo>
        <incarico xml:space="preserve">Direttore</incarico>
      </componente>
    </componenti>
  </organo>
  <organo cod="CLE-SA" cod_amm_aoo="UNKNCLE">
    <nome xml:space="preserve">Senato Accademico</nome>
    <componenti>
      <componente>
        <nominativo xml:space="preserve">Prof. Mario Rossi</nominativo>
        <incarico xml:space="preserve">Presidente</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Cesare Verdi</nominativo>
        <incarico xml:space="preserve">Vicepresidente</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Stefano Stefani</nominativo>
        <incarico xml:space="preserve">Rappresentante Rettore dell'Università degli Studi di Padova</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Leonardo Marroni</nominativo>
        <incarico xml:space="preserve">Rappresentante Rettore dell'Università di Firenze</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Giuseppe Bianchi</nominativo>
        <incarico xml:space="preserve">Rappresentante Rettore dell'Università degli Studi di Napoli, Federico
II</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Prof. Armando Turchesi</nominativo>
        <incarico xml:space="preserve">Rappresentante del Rettore dell'Università di Venezia</incarico>
      </componente>
      <componente>
        <nominativo xml:space="preserve">Dott. Marco Viola</nominativo>
        <incarico xml:space="preserve">Direttore</incarico>
      </componente>
    </componenti>
  </organo>
</Response>

```

Elenco delle sedute di un organo

URL

http://host/titulus_ws/sedute/

Metodo di richiesta supportato

GET

Parametri della chiamata

- **ao** (obbligatorio): rappresenta il codice AMMAOO dell'organo di cui si vogliono le sedute.
- **idorg** (obbligatorio): codice dell'organo di cui si vogliono le sedute.
- **anno** (opzionale): indica l'anno delle sedute.
- **stato** (opzionale): indica lo stato delle sedute che si vogliono recuperare. I valori consentiti sono "aperta", "firma" e "chiusa".
- **soloPubblicabili** (opzionale): indica la volontà di trovare le sedute per cui la data attuale cade all'interno del loro intervallo di pubblicazione (valori "y"|"n"). Questo parametro ha senso solo per le sedute aperte o alla firma, perché quelle chiuse sono sempre pubblicabili.

Esempio di invocazione:

http://host/titulus_ws/sedute/?ao=UNKNCLE&idorg=CLE-CdA

Risposta della chiamata

Di seguito si riporta un esempio di risposta XML.

```
<Response size="3">
  <seduta id="495" cod_amm_ao="UNKNCLE" codOrg="CLE-CdA" data="20130215" straordinaria="si"/>
  <seduta id="496" cod_amm_ao="UNKNCLE" codOrg="CLE-CdA" data="20130325" straordinaria="no"/>
  <seduta id="498" cod_amm_ao="UNKNCLE" codOrg="CLE-CdA" data="20130427" straordinaria="no"/>
</Response>
```



Le date presenti nel xml possono avere i formati GG/MM/AAAA o AAAAMMGG.

Consultazione dell'odg di una seduta

URL

http://host/titulus_ws/odgseduta/

Metodo di richiesta supportato

GET

Parametri della chiamata

- **ao** (obbligatorio): rappresenta il codice AMMAOO dell'organo della seduta.
- **idorg** (obbligatorio): codice dell'organo della seduta.
- **idsed** (obbligatorio): id della seduta.
- **omissis** (opzionale): indica se si vogliono o meno anche le versioni dei file con **omissis** (valori "y"|"n").

Esempio di invocazione:

http://host/titulus_ws/odgseduta/?ao=UNKNCLE&idorg=CLE-CdA&idsed=495

Risposta della chiamata

Gli XML restituiti da questa chiamata sono illustrati nel paragrafo [Esempi di xml di sedute](#). L'unica differenza con questi xml è che l'elemento radice "Response" è privo di attributi.

Fare riferimento anche al paragrafo [Alcune note](#) in cui vengono dati dei dettagli importanti sui file delle comunicazioni/proposte.

Download di un file

URL

http://host/titulus_ws/attach/

Metodo di richiesta supportato

GET

Parametri della chiamata

- **id** (obbligatorio): id del file che si vuole scaricare. Questo id deve essere recuperato dal codice XML dell'odg di una seduta (come indicato in [Alcune note](#)) e non deve essere memorizzato perché può cambiare tra un'invocazione e un'altra dell'url per la consultazione dell'odg di una seduta. **Effettuare sempre l'URL escaping di questo parametro** perché l'id può contenere dei caratteri (es. +) che possono inficiare la chiamata.
- **pdf** (opzionale): indica se si vuole la versione pdf del file (valori "y"|"n"). Qualora non sia disponibile la versione pdf, perché non è stato possibile generarla dinamicamente, viene restituito il file nel formato originale.

Esempio di invocazione:

http://host/titulus_ws/attach/?id=tTpJ/HOJUGSEy6gYjAzHuw==_368.rtf

Risposta della chiamata

La chiamata per il download di un file restituisce un codice XML solo in caso di errore (risposta http 400/500). Qualora non si verifichi alcun errore, viene restituito direttamente il contenuto del file e negli header della risposta http è presente il mime type del file.