

Progetto di fine corso

La valutazione di fine corso per ogni studente è soggetta alla realizzazione di un progetto che dimostri le conoscenze e le abilità acquisite nell'uso delle tecnologie per lo sviluppo di Web API, stimolando anche l'approfondimento di argomenti avanzati non trattati in aula.

- I progetti possono essere realizzati singolarmente o in piccoli gruppi. Ovviamente il numero di persone coinvolte in un lavoro aumenterà le richieste di completezza e correttezza dello stesso.
- Il contributo effettivo offerto da ciascun componente di un gruppo alla realizzazione finale dovrà essere descritto nella documentazione. In sede di esame, i responsabili potranno essere chiamati a riferire sugli aspetti loro delegati.

Introduzione

Il progetto di quest'anno vi porterà ad esplorare il temibile mondo dei *web services scritti male*, e richiederà un uso "non standard" di tutto quello che avete imparato su SOAP e WSDL. Lavoreremo su un vero web service SOAP, esposto dalla segreteria virtuale della nostra Università.

Prima di tutto, quindi, una **raccomandazione**: ricordate che questo è un sistema reale, in esecuzione presso il nostro centro di calcolo, e gestisce le carriere di tutti gli studenti (voi compresi). Di conseguenza non ci "giocate" troppo, evitate i flooding di richieste, e non mandate in giro questa URL. Sebbene le API importanti siano protette da autenticazione, non è comunque il caso di abusare di quelle pubbliche, su cui invece lavoreremo.

Lo **scopo** di questo progetto è semplice: *trasformare questa incomprensibile e poco usabile API SOAP in una API RESTful più agile e moderna*. Come? Scrivendo un nuovo servizio RESTful che espone una API *fatta bene* e che, dietro le quinte, chiama la API SOAP *fatta male* in maniera opportuna e ne decodifica i risultati. Insomma, un servizio "gateway" tra due mondi diversi.

In **allegato** a questa specifica c'è un documento di analisi funzionale del "WEB SERVICES ESSE3" su cui ho avuto modo di mettere le mani tempo addietro. Anche se questo è un documento-guida ufficiale, riguarda un prodotto commerciale, quindi *usatelo internamente e non distribuitelo all'esterno in nessun modo*.

Le **operazioni** su cui dovrete concentrarvi sono quelle della prima parte "**Area Didattica – Elenco delle Operazioni**". Tutte le altre potrebbero non funzionare o non restituire dati. Siete liberi di lavorarci sopra e provare a trarre qualcosa anche dalle altre operazioni, ma non è necessario. In ogni caso, anche se riuscite ad avere accesso alle API protette dalla login (magari con le vostre credenziali della segreteria virtuale) **NON** provate a effettuare operazioni in scrittura, perché queste andrebbero veramente a modificare il vostro profilo studente!

Ovviamente questo lavoro lo scrivente l'ha già fatto, quindi sa cosa significa... e vi darà una mano a iniziare.

Come Procedere

Il servizio è disponibile all'indirizzo <https://segreteriavirtuale.univaq.it/services/ESSE3WS>, e ovviamente il suo WSDL è scaricabile con la URL <https://segreteriavirtuale.univaq.it/services/ESSE3WS?wsdl>. Leggetelo, e capirete quello che intendo per “scritto male”: a parte un paio di operazioni, le altre sono veramente criptiche.

Per prima cosa, dimenticatevi di creare un client “automatico” usando JAX-WS, perché questo WSDL usa la temibile combinazione *RPC/encoded* (che infatti vi ho sconsigliato a lezione), che JAX-WS non riesce a interpretare bene. Dovrete quindi chiamare il servizio a mano, creando i relativi messaggi SOAP. Non è difficile: ricordate solo che la chiamata SOAP-RPC non è altro che una POST verso la URL del servizio tra i cui header c'è una SOAPAction e nel cui payload c'è un documento SOAP, appunto. In ogni caso, come vedremo, riuscire a chiamare le operazioni descritte nel WSDL non vuol dire, in questo caso, poter subito sfruttare la API che ci serve.

Dove sono infatti, nel WSDL, tutte le operazioni elencate nel documento di analisi funzionale allegato? Semplice, non ci sono. Quelle operazioni sono una sub-API a cui si accede usando una sola operazione del servizio: **fn_retrieve_xml_p**. Analizzate il WSDL e cercate di capire come funziona. Usando un tool come SOAPUI potreste farvi generare automaticamente il messaggio SOAP che serve a chiamarla, che per praticità vi riporto qui:

```
<soapenv:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://ws.esse3.frk.kion.it">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:fn_retrieve_xml_p
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <retrieve xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">CODICE_OPERAZIONE</retrieve>
      <params xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">PARAMETRI_DI_INPUT</params>
      <xml xsi:type="soapenc:string" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" />
    </ws:fn_retrieve_xml_p>
  </soapenv:Body>
</soapenv:Envelope>
```

Quindi, se inviate in POST la struttura di cui sopra all'indirizzo <https://segreteriavirtuale.univaq.it/services/ESSE3WS>, avendo cura di aggiungere un header *SOAPAction* con valore vuoto (come richiesto nel binding del WSDL), riceverete la vostra risposta, che ovviamente dipende da quello che avete sostituito nel messaggio a *CODICE_OPERAZIONE* e *PARAMETRI_DI_INPUT*. Entrambi questi valori corrispondono a quelli che trovate nella specifica funzionale. Attenzione: da quel che sono riuscito a capire, i nomi dei parametri e delle operazioni sono sempre *case sensitive* e non sono ammessi spazi bianchi all'interno o prima di queste stringhe.

In particolare, i parametri devono essere passati nella forma *nome=valore* e, se ne sono richiesti più di uno, vanno separati da un punto e virgola, ad esempio *aa_ord_id=2015;docente_matricola=003088*. Se un parametro ha come valore una stringa, dovete metterlo tra apici, ad esempio *tipo_corso='LM'*.

Quindi, se volessi chiamare la funzione “Anno accademico corrente” specificando “Calcolo AA nel Calendario Esami” come modalità, dovrei mandare questo messaggio

```
<soapenv:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ws="http://ws.esse3.frk.kion.it">
  <soapenv:Header/>
  <soapenv:Body>
    <ws:fn_retrieve_xml_p soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <retrieve xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">GET_CURR_AA</retrieve>
      <params xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">tipo_data_rif_cod=DR_CALESA</params>
      <xml xsi:type="soapenc:string" xmlns:soapenc=http://schemas.xmlsoap.org/soap/encoding/" />
    </ws:fn_retrieve_xml_p>
  </soapenv:Body>
</soapenv:Envelope>
```

La struttura delle risposte dovete capirla voi, ma è molto semplice, basta estrarne quello che vi serve e lasciar stare il resto. Nel caso dell'esempio riceverete questa risposta:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:fn_retrieve_xml_pResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://ws.esse3.frk.kion.it">
      <fn_retrieve_xml_pReturn href="#id0"/>
      <xml xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <![CDATA[
          <?xml version="1.0" encoding="Windows-1252"?>
          <WS>
            <DataSet LocalEntityName="AA_CORRENTE">
              <Row Num="1">
                <aa_id>2019</aa_id>
                <data_inizio>01/10/2019 00:00:00</data_inizio>
                <data_fine>30/09/2020 23:59:59</data_fine>
                <des>2019/2020</des>
              </Row>
            </DataSet>
          </WS>
        ]]>
      </xml>
    </ns1:fn_retrieve_xml_pResponse>
    <multiRef id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="xsd:int"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">1</multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

Se vedete delle risposte vuote (SOAP Body senza dati ma con solo dei codici di ritorno), vuol dire che state sbagliando qualcosa nella richiesta.

Suggerimenti

Nel provare le operazioni, tenete presente che in generale **vanno usate dall'alto verso il basso**, perché molte hanno bisogno di parametri (codici, id e simili) restituiti nelle risposte di operazioni precedenti. Quindi, ad esempio, se vedete che è richiesto un parametro che si chiama *aa_id* (o *aa_ini_id...*), andate a vedere se prima qualche operazione dell'API vi può restituire un omonimo *aa_id* tra i dati della risposta, per capire i valori che potete utilizzare.

Esplorate prima la API con SOAPUI e/o con Postman, e solo alla fine scrivete il codice che effettua le chiamate, trasformandolo in un'API Java e poi in un'API RESTful.

Come al solito, attenti a usare bene la struttura delle URL, perché c'è sicuramente una *gerarchia* in questa API, mentre per i verbi in questo caso c'è poco da fare, sarà praticamente tutto GET.

Cosa Consegnare

Quello che mi aspetto da voi è un servizio funzionante, installato su una vostra macchina privata (niente online, mi raccomando!), che possiate interrogare con Postman per mostrarne interattivamente il funzionamento. Ovviamente vedremo insieme anche come avete scritto il codice.

Da buon servizio RESTful, non sarà auto-documentato, quindi dovrete scrivere una documentazione in cui illustrate le API da voi realizzate elencando le relative URL, i verbi applicabili su ciascuna di esse, i dati in input e output e i codici di stato restituiti, oltre ovviamente a una breve descrizione dell'operazione svolta.

Come verrete valutati

Quello che valuterò nei vostri progetti sarà la buona definizione della API RESTful in astratto, poi il modo in cui è stata implementata. Valuterò inoltre come e quanto siete riusciti a interfacciarvi con la API SOAP di origine, nonché quanto di quella API siete riusciti a sfruttare e “mappare” sulla vostra RESTful. Infine, ovviamente prenderò in considerazione anche la qualità del codice in generale.