AGENZIA DELLE DOGANE
E DEI MONOPOLI

Web Services Dogane

LINEE GUIDA

Indice

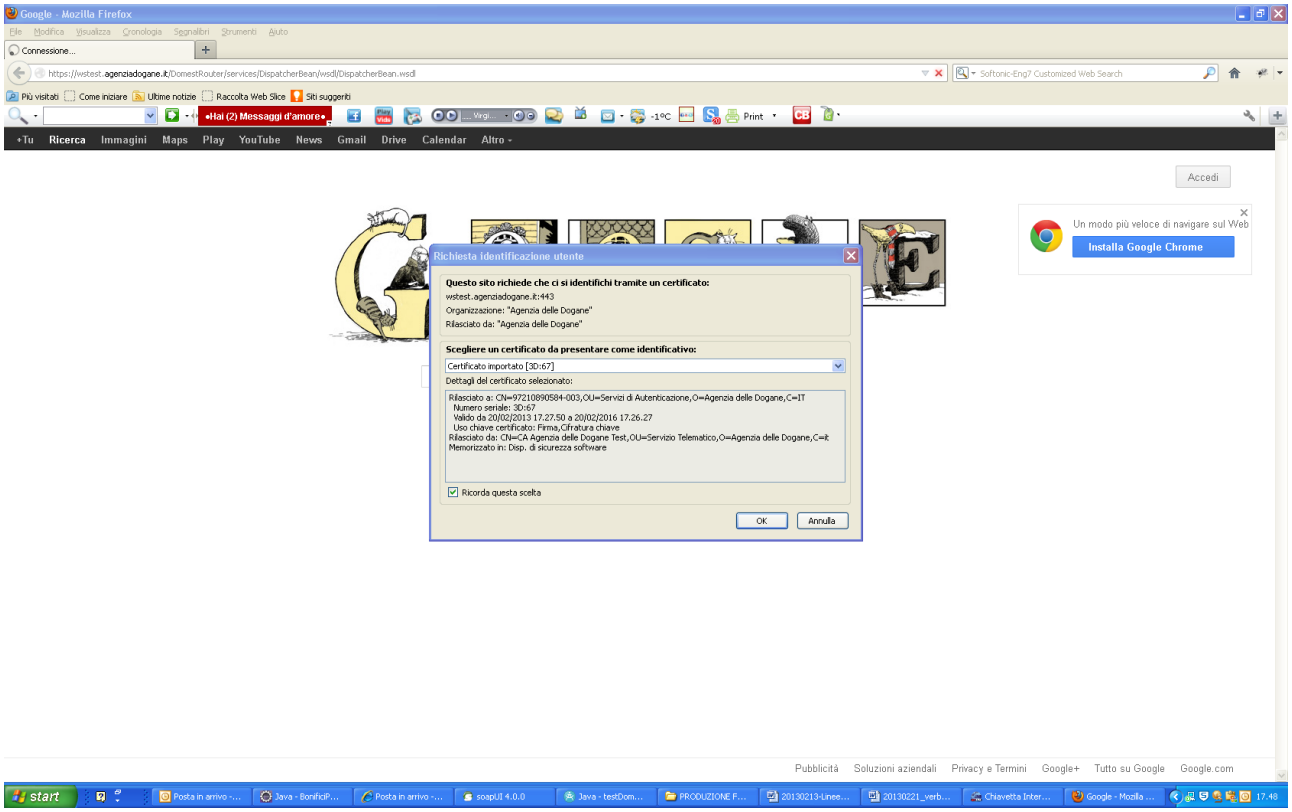
1. INTRODUZIONE

Un Web Service è per definizione un servizio descritto dal fornitore (provider) in modo standard e indipendente dal linguaggio di programmazione in cui è stato sviluppato.
Per usufruire dei Web Services forniti dal provider (Agenzia delle Dogane e dei Monopoli) occorre creare un client di Web Services.
Ciò è possibile attraverso il file WSDL (Web Services Description Language) che definisce l'interfaccia del servizio, cioè l'elenco delle operazioni fornite, i vari parametri che ogni operazione si aspetta di ricevere da chi la richiama e l'elenco dei parametri che l'operazione fornisce come output.
Un Web Service viene richiamato parametricamente dall'utente del servizio (richiedente) attraverso programmi applicativi e basandosi su un protocollo di chiamata remota, indipendente dalla rete e dai linguaggi di programmazione, detto SOAP (Simple Object Access Protocol).
Un documento WSDL è un documento XML che contiene un insieme di definizioni per descrivere il servizio (Web Service). Gli elementi più importanti utilizzati da WSDL sono i seguenti:

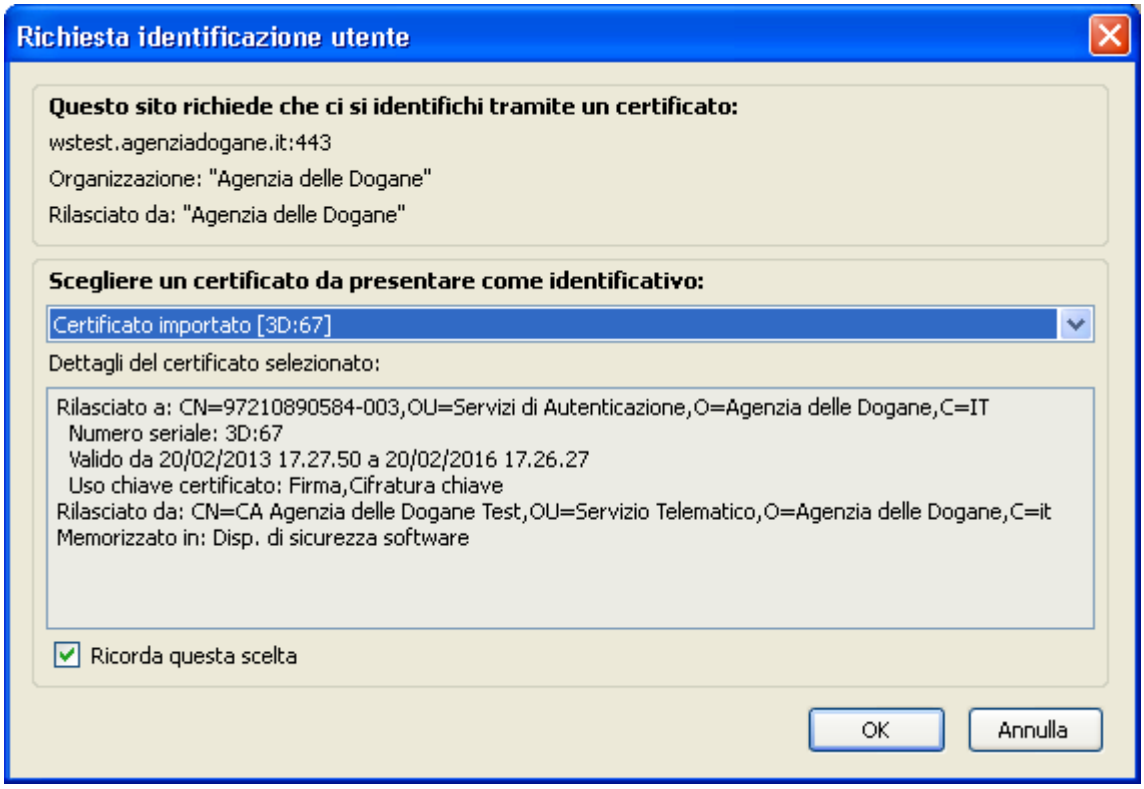
- **<types>**
  definisce i tipi di dati utilizzati nel servizio;
- **<message>**
  contiene le definizioni dei messaggi di scambio del servizio utilizzando parti definite come tipi nella sezione types;
- **<portType>**
  descrive il servizio, le operazioni che possono essere eseguite e i messaggi che sono coinvolti in queste operazioni. Per ogni metodo viene definito il messaggio di input ed il messaggio di output. L'elemento porType può essere paragonato ad una libreria di funzioni in un tradizionale linguaggio di programmazione;
- **<binding>**
  contiene il collegamento tra il portType (cioè la definizione astratta del servizio) e l'end-point fisico. Questa informazione indica il protocollo da utilizzare e come ricondurre i messaggi di input ed output al protocollo utilizzato;
- **<port>**
  definisce la porta di accesso al servizio .Un servizio può avere anche più porte ciascuna con un nome e un protocollo di binding;
- **<service>**
  contiene la definizione del servizio in termini della sua descrizione e della posizione fisica del servizio (tipicamente il suo URL) – definiti endpoint.

Il file WSDL che descrive i servizi (Web Services) esposti dall'Agenzia delle Dogane e dei Monopoli è disponibile sul sito dell'Agenzia delle Dogane e dei Monopoli nella sezione web service (Ambiente reale e Ambiente di prova) dedicata allo specifico servizio. Nel file è altresì indicato l'endpoint del servizio stesso.

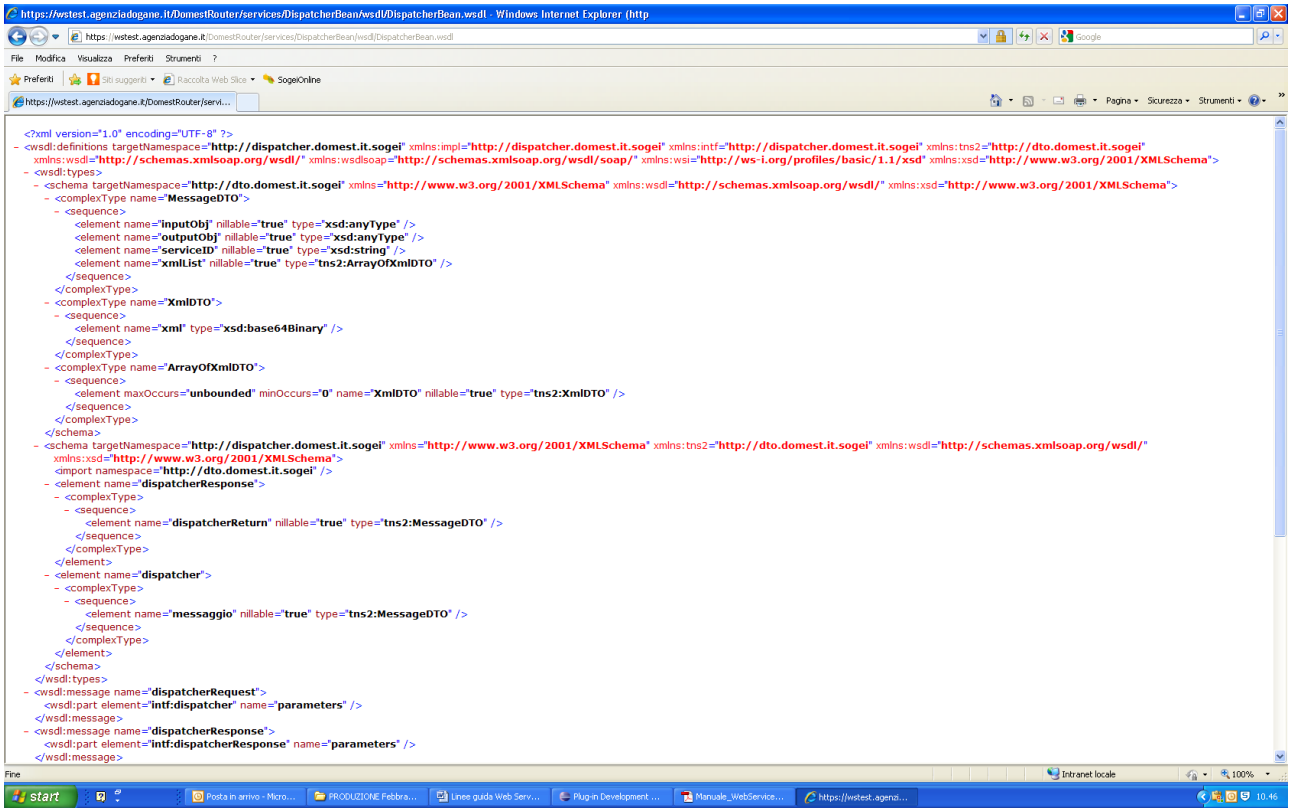
Per poter accedere all'endpoint del servizio è necessario caricare il certificato di autenticazione nel proprio browser (la procedura di importazione dipende dal browser utilizzato). Digitando la URL viene infatti visualizzata la seguente maschera di scelta del certificato da utilizzare:



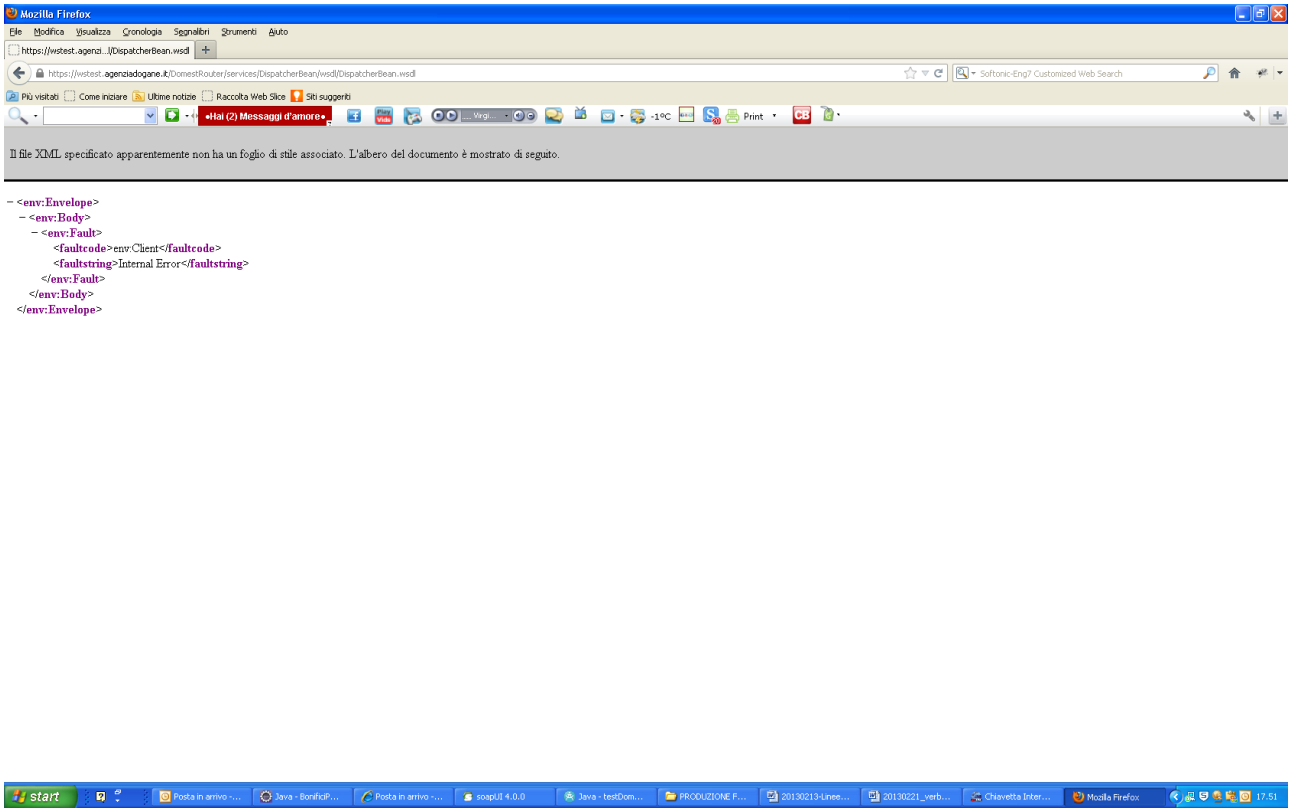
Scegliere tra quelli disponibili, il certificato precedentemente caricato.



Una volta effettuato l'accesso con il certificato (senza più bisogno di username e password), concatenando all'endpoint il valore ?wsdl, il sistema dovrebbe fornire in risposta il WSDL dell'applicazione. Per motivi di sicurezza gli apparati di governance utilizzati potrebbero impedire le operazioni di GET per il protocollo HTTP, negando di fatto la possibilità di scaricare il wsdl.



Come detto, per motivi di sicurezza, al posto della schermata precedente con il wsdl, potrebbe presentarsi la pagina seguente, che rappresenta comunque un test di connettività soddisfacente.



Utilizzando taluni browser (Microsoft Internet Explorer per esempio) è possibile che venga restituito un altro tipo di errore. Sebbene la richiesta da parte del server di sottomettere il proprio certificato di autenticazione già rappresenti di per sé un test di connettività soddisfacente, per assicurarsi di aver superato il test di connettività si consiglia l'utilizzo dello strumento open SoapUi o similare, come descritto nel capitolo successivo in merito ai test funzionali.

2. TEST FUNZIONALI SUI WEB SERVICES

E' consigliabile effettuare un test sui Web Services esposti prima della creazione del client vero e proprio. A tale scopo l'utente si può avvalere ad esempio del tool SoapUI, uno strumento Java Open Source, rilasciato sotto la licenza GNU LGPL (Lesser General Public License), estremamente utile e utilizzabile su qualsiasi piattaforma (Windows/Unix/Linux).
SoapUI è destinato a sviluppatori e collaudatori di servizi web in quanto consente di ispezionare i servizi web, richiamarli, realizzarli e fare delle prove di carico.
I test funzionali e di caricamento possono essere fatti sia interattivamente, usando una comoda interfaccia utente, che attraverso un processo automatico, grazie all'uso dei tool a linea di comando.
Per quanto riguarda le funzionalità di invocazione e analisi dei Web Services, SoapUI mette a disposizione:

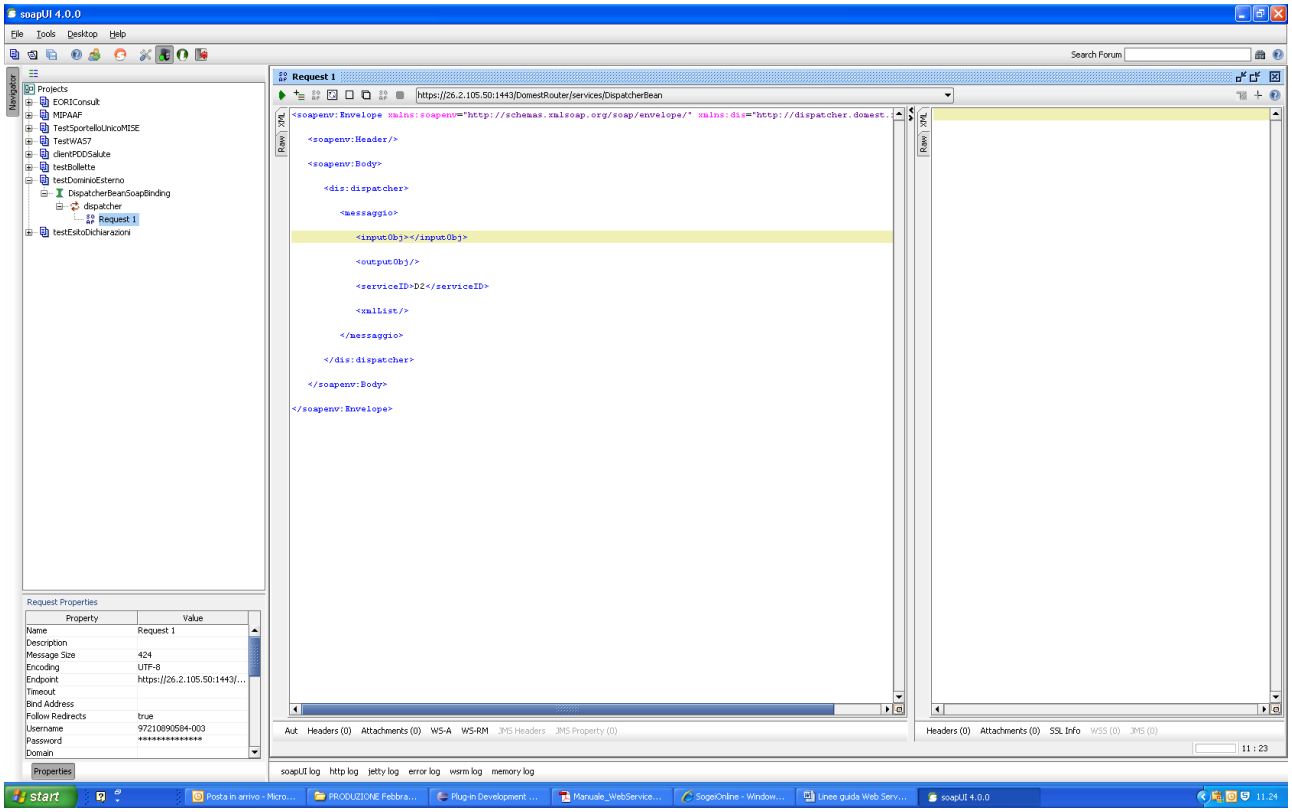
- importazione del WSDL;
- generazione automatica delle richieste;
- supporto per i vari tipi di autenticazione (Digest, WS-Security, NTLM, ecc.);
- supporto di SOAP 1.1 e 1.2;
- editor con sintassi colorata e funzionalità di undo/redo e formattazione automatica.

Per lo sviluppo e la validazione dei Web Services vi sono:

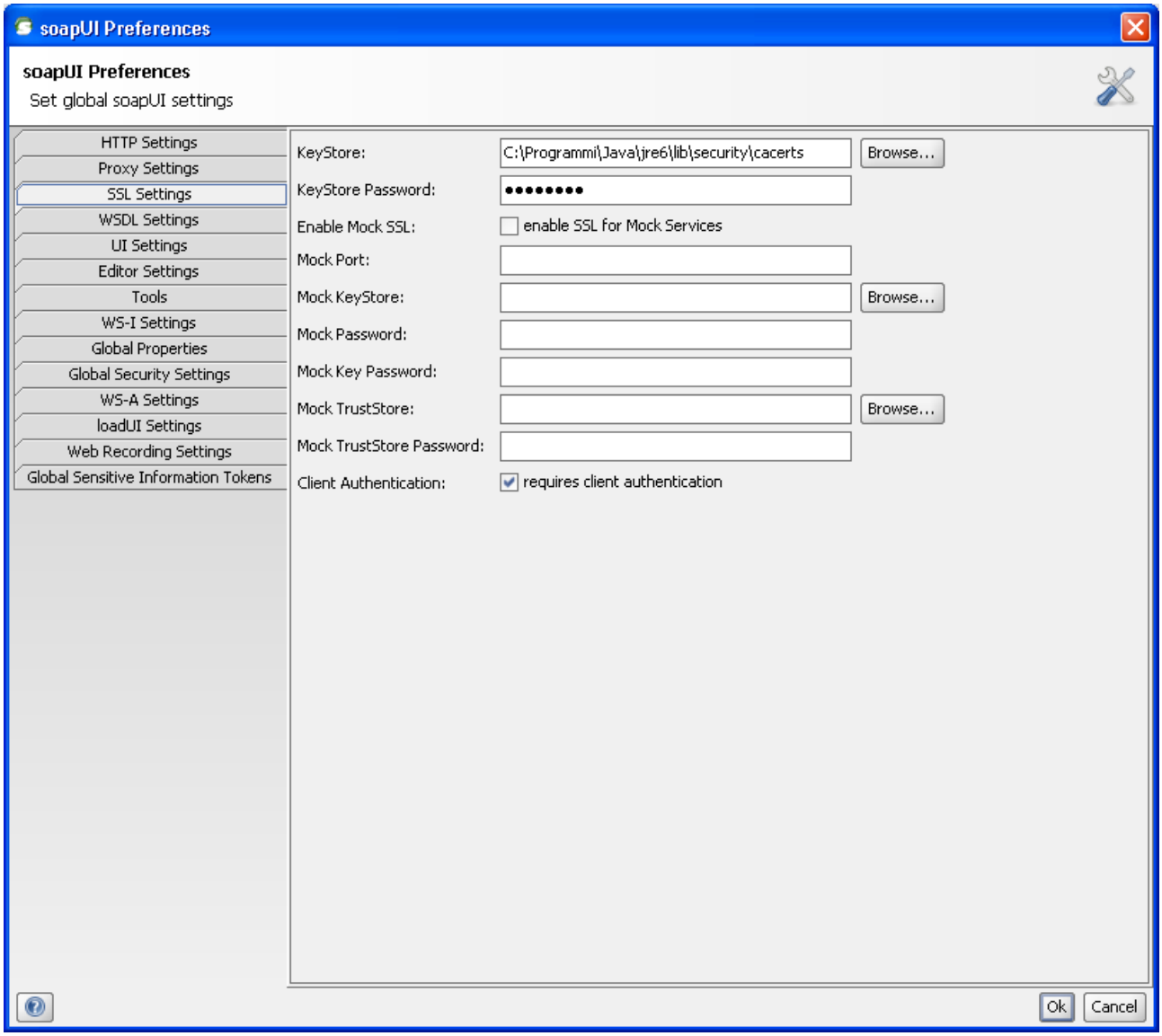
- generazione del codice sia Client che Server con supporto dei toolkit più diffusi: JBossWS, JWSDP, Axis 1 e 2, XFire, .NET e gSOAP;
- generazione del WSDL da codice Java esistente;
- generazione delle classi di binding XML per JAXB e XMLBeans;
- validazione delle definizioni dei Web Services.

Inoltre è possibile usare potenti funzioni per quanto riguarda sia i test funzionali dei Web Service che le prove di carico. In questo caso è possibile avere report dettagliati, statistiche varie, log completi e analisi sulle prestazioni.

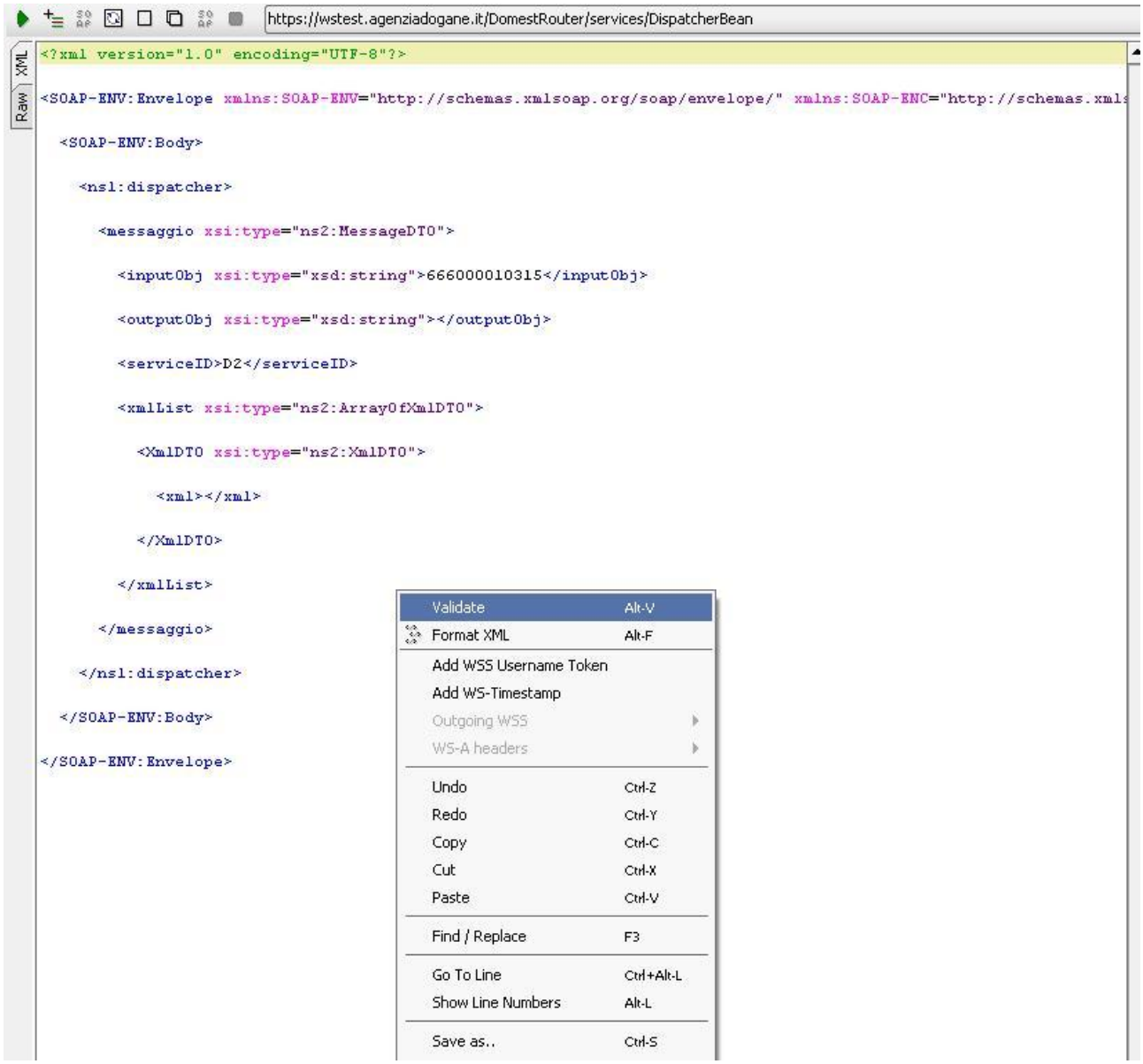
L'interfaccia dell'applicazione è intuitiva e facile da utilizzare come mostrato nella figura riportata di seguito.



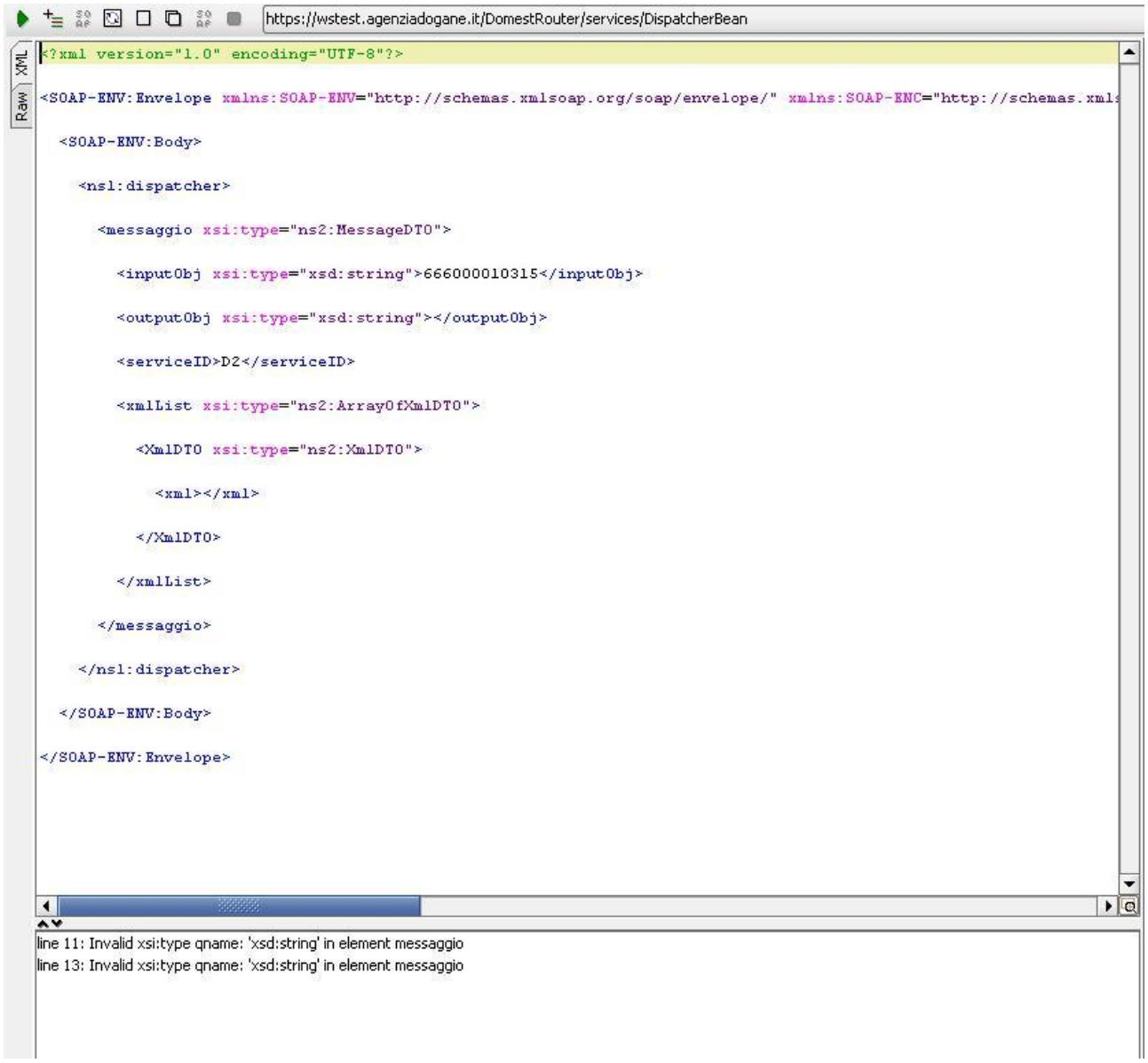
Dal menu File, selezionare la voce Preferences e nella schermata successiva scegliere la voce “SSL Settings”. Indicare il proprio certificato client di autenticazione con la relativa password. Spuntare *requires client authentication* e premere ok. Nel menu File cliccare sulla voce Save Preferences se presente.



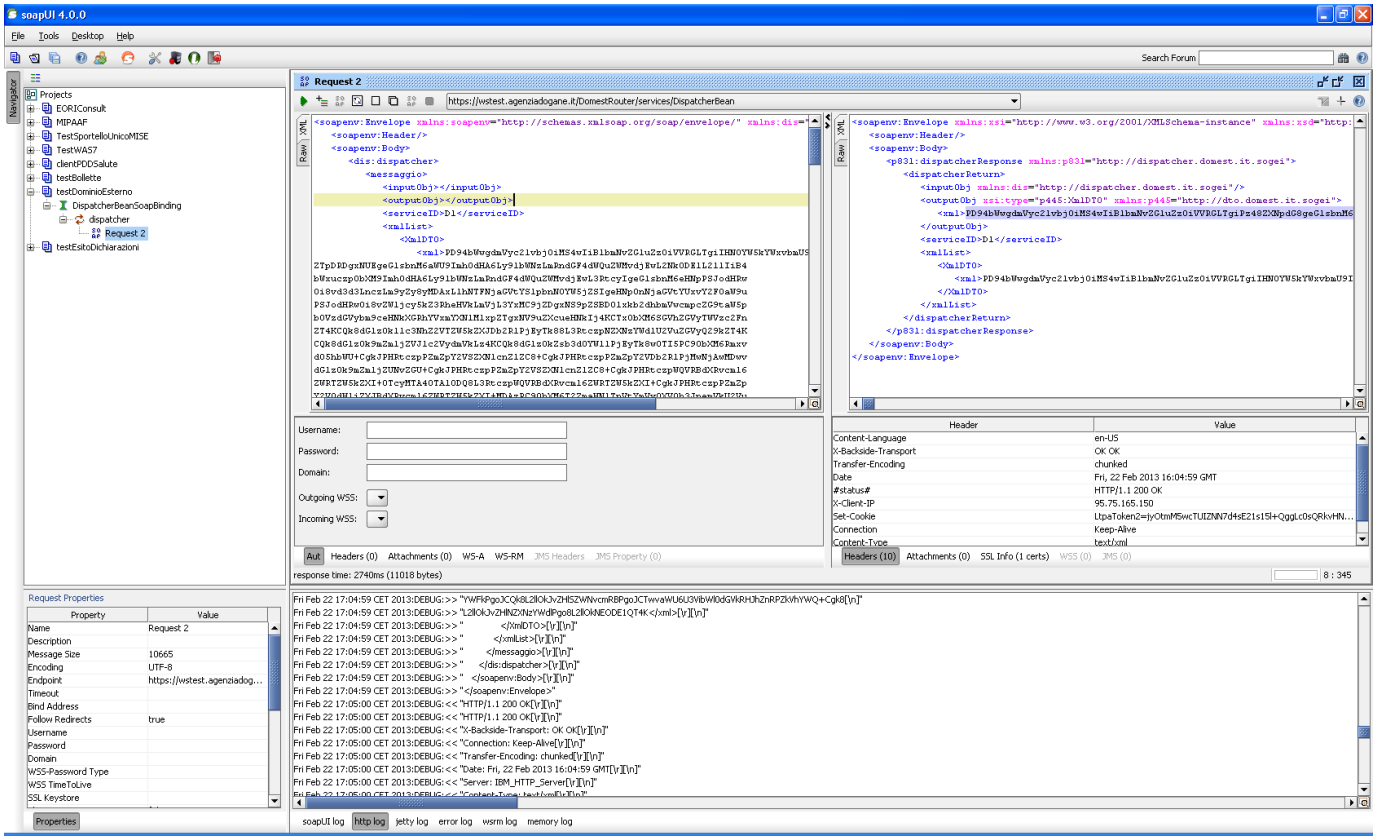
Si ritiene opportuno effettuare la validazione della request prima di proseguire. I due screenshot successivi mostrano come accedere con il tasto destro del mouse alla validazione della request.



Di seguito un esempio di request sbagliata con indicazione del problema.



Una volta compilata correttamente la request, inserendo obbligatoriamente il serviceID e i parametri indicati sul manuale, si ottiene una risposta dal sistema (vedi figura seguente).


soapUI 4.0.0
File Tools Desktop Help
Search Forum
Navigator
Projects
EORIConsult
MIPAAF
TestSportelloUnicoMISE
TestWAS7
clientPDDSalute
testBollette
testDominioEsterno
DispatcherBeanSoapBinding
dispatcher
Request 2
testEsitoDichiarazioni
Request 2
https://wstest.agenziadogane.it/DomestRouter/services/DispatcherBean
Username:
Password:
Domain:
Outgoing WSS:
Incoming WSS:
Aut Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0)
response time: 2740ms (11018 bytes)
Header
Value
Content-Language en-US
X-Backside-Transport OK OK
Transfer-Encoding chunked
Date Fri, 22 Feb 2013 16:04:59 GMT
#status# HTTP/1.1 200 OK
X-Client-IP 95.75.165.150
Set-Cookie LtpaToken2=jyOtmM5wcTUIZNN7d4sE21s1SH+QggLcOsQRkvHN...
Connection Keep-Alive
Content-Type text/xml
Headers (10) Attachments (0) SSL Info (1 certs) WSS (0) JMS (0)
Request Properties
Property
Value
Name Request 2
Description
Message Size 10665
Encoding UTF-8
Endpoint https://wstest.agenziadog...
Timeout
Bind Address
Follow Redirects true
Username
Password
Domain
WSS-Password Type
WSS TimeToLive
SSL Keystore
Properties
soapUI log http log jetty log error log wsrm log memory log

3. SICUREZZA

L'accesso ai Web Services è regolato da un'autenticazione forte che prevede un certificato di autenticazione rilasciato dal STD agli utenti registrati che ne facciano richiesta.
Di seguito forniamo un esempio di una chiamata dell'operazione dispatcher realizzata in linguaggio java:

```
System.setProperty("java.protocol.handler.pkgs", "com.ibm.net.ssl.internal.www.protocol");
System.setProperty("javax.net.ssl.keyStore", certificatoAutenticazione);
System.setProperty("javax.net.ssl.keyStorePassword", "keyStorePassword");
System.setProperty("javax.net.ssl.keyStoreType", "pkcs12");
System.setProperty("javax.net.ssl.trustStore", truststoreCertificatoServer);
System.setProperty("javax.net.ssl.trustStorePassword", "trustStorePwd");
System.setProperty("javax.net.ssl.trustStoreType", "JKS");
DispatcherBeanProxy proxy = new DispatcherBeanProxy();
Stub stub = (Stub) proxy.getDispatcherBean();
MessageDTO response = proxy.dispatcher(message);
```

4. FIRMA

Il messaggio xml trasferito come byte[] deve essere firmato con XML Digital Signature. Gli standard presi come riferimento sono:

a. XML Advanced Electronic Signature (XAdES) standard, ETSI TS 101 903
b. "XML-Signature Syntax and Processing - W3C Recommendation 12 February 2002"

Si ritiene necessario introdurre i seguenti requisiti tecnici:

- la firma XML è di tipo Enveloped dove l'elemento caratterizzante la firma digitale (ds:Signature) sarà posto come ultimo elemento della radice della struttura XML. Tale documento viene firmato digitalmente tramite l'utilizzo di chiavi e relativo certificato di firma a disposizione dell'operatore.

- uso obbligatorio dell'attributo Id per i tag <ds:Signature>, e <ds:SignatureValue>

In particolare si fa riferimento alle regole tecniche definite dalla DELIBERAZIONE N. 45 DEL 21 MAGGIO 2009 (reperibile su http://www.digitpa.gov.it/firme-elettroniche-certificatori).
Delle tre tipologie di firma xml citate nella deliberazione è necessario che il client di firma generi firme di tipo XAdES-BES enveloped.

Di seguito uno snapshot di codice Java per la firma, a titolo di esempio. In particolare l'esempio sfrutta la JRE 1.5 e le librerie BouncyCastle.

```
public static byte[] sign(byte[] xmlnotsigned) throws Exception {

        byte[] xmlsigned = null;

        //Scrivo file su disco
        OutputStream out = new FileOutputStream(new File("temp.xml"));

        out.write(xmlnotsigned);
        out.close();


        initEnvNew IE = new initEnvNew();
        IE.loadEnv("/conf/Bouncy.properties");

        Vault vault = new Vault();
        KeyStore ks = KeyStore.getInstance("PKCS12");

        ks.load(new FileInputStream(new File("certificatoFirma.p12")),
        pwdCertFirma.toCharArray());

        vault.Open("", ks, pwdCertFirma);

        Provider provider = Environment.setXmlDsigProvider();

        Signer sign = new Signer(provider);
        sign.sign("temp.xml", "tempsigned.xml", "", vault, "Signature1", "SignatureValue1",
        "SignedProperties_1");
        xmlsigned = getBytesFromFile(new File("tempsigned.xml"));

        return xmlsigned;

    }
```

5. TRASFORMAZIONE CERTIFICATO DI FIRMA

Per poter adempiere agli obblighi di firma digitale descritti nel paragrafo precedente, è necessario possedere un certificato PKCS12 per la firma.
Gli utenti registrati al STD e già in possesso del certificato di firma con estensione .ks dovranno effettuare una conversione di tale certificato in un formato coerente con le specifiche, producendo un file con estensione .p12.

Per maggiori dettagli consultare le FAQ relative ai Web Service sul sito dell'Assistenza online http://assistenza.agenziadogane.it.

6. WSDL

I WSDL sono disponibili nelle sezioni del sito:

Web service - Ambiente reale

Web service – Ambiente di prova