

Dokumendi eesmärk

Käesoleva dokumendi eesmärgiks on anda ülevaade DigiDocService testide automatiseerimiseks kasutatud vahendist soapUI. Kirjeldatakse projektifaili ja projekti ülesehitust, testikomplektide ja testistsenaariumide koostamist, testide käivitamist käsurealt ning tulemuste failidesse suunamist.

Seejärel kirjeldatakse, kuidas koostatud DigiDocService testides orienteeruda ning kuidas käivitada vajalikke teste ning saada teada nende tulemusi.

Testide automatiseerimise vahend soapUI

SoapUI on javapõhine veebiteenuste testimisvahend sooritamaks funktsionaalseid ja koormusteste. Tarkvara tootjaks on Rootsi firma Eviware. SOA Web Services Journali käimasolevas lugejaküsitluses on see tööriist praegu paremuselt teine nii veebiteenuste testimise kui ka parima veebiteenuste vahendi kategoorias¹. Tarkvara esimene versioon ilmus 2005 aasta septembris. SoapUI on litsentseeritud LGPL litsentsi alusel.

Testimist saab läbi viia interaktiivselt soapUI graafilisest kasutajaliidesest või command-line tööriistade abil. SoapUI eeldab java 1.5 –te. Praeguse projekti raames on kasutatud soapUI versiooni 1.6 kuid lähinädalatel saab kättesaadavaks uus versioon 1.7.

SoapUI projekti struktuur

SoapUI projektifail on XML kujul, mis võimaldab lihtsamaid parameetrite asendusi teha mõne XML editoriga. Projektifail sisaldab:

1. veebiteenuse kirjeldust (WSDL),
2. veebiteenuse lõpp-punkte,
3. testikomplekte
4. neis sisalduvaid erinevaid testistsenaariume.

Sama projektifaili alusel on teste võimalik käivitada ka command-line režiimis.

SoapUI projekti ekraanipilt on toodud joonisel 1.

Projekt on üles ehitatud hierarhilise puuna (ekraani vasakpoolne jaotis), kus üks haru kajastab veebiteenuse liidest ja kõikide veebipäringute põhjasid. Neid

¹ <http://soa.sys-con.com/general/readerschoice.htm>

veebipäringute põhjasid saab täiendada oma soovide kohaselt kasutades kloonimist ja erinevaid lähteparameetreid. Lisaks saab veebiteenuse juures defineerida veebiteenuste lõpp-punktid, kuhu oma päringuid saata.

Projekti teised harud kajastavad testkomplekte (*TestSuite*), mis omakorda hõlmavad mitmeid sarnaseid teststsenaariume (*TestCase*, testilugu). Iga testkomplekti kõiki teststsenaariume saab käivitada paralleelselt või järjestikku (valitakse testkomplektil topeltklikiga avaneval ekraanil, kus kuvatakse testkomplekti kuuluvad teststsenaariumid). Samuti saab käivitamiseks valida ainult mõned testkomplektis olevad stsenaariumid. Iga teststsenaariumi saab omakorda käivitada ühekordsena või pidevana (määratakse teststsenaariumil topeltklikiga avaneval ekraanil, kus kuvatakse teststsenaariumi kuuluvad testisammud). Ka on võimalik testisamme üksikult käivitada (testisammul topeltklikiga avanevalt ekraanilt).

Lisaks on teststsenaariumi ning testisammu ekraanil võimalik määratata eraldi teenuse lõpp-punkt. Command-line režiimis on teenuse lõpp-punkt parameetrina etteantav.

Testistsenaarium koosneb erinevatest testisammudest mille liikideks on:

- *Properties* globaalmuutujate hoidmine ühe stsenaariumi piires. Põhiline kasutusala sessiooni kood või DigiDoc fail mõnes tekstimuutujas.
- *Property Transfer* Andmete liigutamiseks eri sammude vahel
- *Delay* Viite saavutamiseks sammude vahel (millisekundites)
- *Goto* Tingimuslik suunamine erinevate sammude vahel. Põhiliselt kasutaja tegevuse pollimiseks meetodite `GetStatusInfo` või `GetMobileAuthenticateStatus` juures. Tingimus peab olema vormistatud XPath avaldisena.
- *Request* Parameetritega päring veebiteenusele. Tavaliselt kirjas meetodi nimega. Selle sammu juures saab päringu vastusele peale ehitada erinevaid veakontrolle.
- *Groovy Script* Samm, kus peaks saama teha kõike. Skript kompileeritakse dünaamiliselt Java baitkoodiks. Põhiliselt kasutusel tsükli tegemiseks üle erinevate DigiDoc failide, et neid faile verifitseerida. Näiteks on võimalik külge haakida JDBC draiverid, mida siis andmete sisse- või väljalugemiseks kasutada.
- *Property Expansion* Eeldusel, et sammu nimi jääb samaks, saab andmeid ühest sammust teise ka tõsta kujul `${Sammunimi#Väljanimi}`. Sellisel juhul kaob ära vajadus Property Transferi järele, kuid siis teksti kodeerimist ei tehta. Leiab kasutust sessiooni koodi laiali jagamiseks järgnevate päringute vahel.



www.sk.ee

DigiDocService automaattestid

Kasutusjuhend

Versioon 1.0 – 13.03.2009

Et kontrollida, kas testistsenaarium on läbitud ootuspäraselt või mitte, on võimalik kirjeldada ka validatsioonitegevusi (*Assertion*, veakontroll).

Vastussõnumitele rakendatavate veakontrollide liigid:

- *Schema Compliance* Kahjuks soapUI ei toeta SOAP-Encoded sõnumite valideerimist. Seetõttu ei kasutata.
- *Simple Contains* Stringi otsimiseks vastussõnumist. Kui ei leia, annab vea.
- *XPath Match* Kui XPath avaldis annab false, siis antakse viga. Kasutatakse põhiliselt XML tag -i otsimiseks vastussõnumist.
- *Soap Fault Assertion* Võrdleb vastussõnumit teenuse kirjeldusega. Kui muster ei klapi, annab vea.
- *Simple NotContains* Stringi, mida ei tohiks vastussõnumis eksisteerida, otsimiseks vastussõnumist. Kui ei sisaldub, annab vea.

Testijuhu käivitamisel interaktiivselt kuvatakse kohe ka testide käivitamise logi (mis sammu juures parajasti ollakse ning kui kaua mingi samm aega võttis) ning testisammude tulemused (OK või FAIL) on vaadeldavad avades testisammu ekraani.

Testide käivitamine käsurealt

Testrunner on command-line vahend käivitamiseks juba varem valmis tehtud teste.

Kasutatavad parameetrid:

- e teenuse lõpp-punkti määramine, et sõita üle projektifailis määratu.
- f kataloogi määramine testi tulemi maha kirjutamiseks
- s testkomplekti valik juhul kui ei soovita käivitada kogu projekti teste.
- c käivitamiseks ühe testistsenaariumi valimine
- r genereerib tulemist väikese kokkuvõtte
- a ekspordib ka korrektsed tulemused väljundfailidesse

Näide: testrunner.bat -fD:\tulem3 -r -a D:\soapui-project.xml
-s "Kiipkaardiga allkirjastamise testid"
-c "TC_ALLK_KIIP_01"
-e https://www.sk.ee:8097/DigiDocService

Tulemusfailid tekkivad etteantud kataloogi järgneval nimekujul.

```
<testkomplekti_nimi>-  
  <teststenaariumi nimi>-  
    <sammu nimi>-  
      <sammu järjekorranumber>-  
        <sammu staatus>.txt
```

Näide:

DigiDoc failide verifitseerimise testid-TC_VERIF_04-StartSession koos ddoc failiga-0-FAILED.txt

DigiDocService automaattestide kasutamine

Testikomplektide loetelu on koostatud vastavalt testilugude teemadele:

- DigiDoc failide verifitseerimise testid
- Kiipkaardiga allkirjastamise testid
- Kiipkaardiga allkirjastamise robustness testid
- Koormustestid

Neid testilugusid saab käivitada soapUI keskkonnas interaktiivselt või käsurealt vastavalt ülevalpool toodud näitele.

Testilugude nimekuju testprojekti langeb kokku üldjuhul kokku testilugudes märgituga. Mõned testilugude dokumendis kirjeldatud testilood on tükeldatud erinevate teststenaariumite vahel projektis. Sellisel juhul langeb nimekuju esimene pool projektis kokku nimega testilugudes.

DigiDoc testfailid ja parameetrifailid asuvad **testifailid** kataloogis ja seda projektifaili asukoha suhtes. Parameetrifaili nimekuju *TC_VERIF_04.txt* on analoogne testilugudes antud kodeeringuga. Parameetrifail sisaldab nende DigiDoc failide loetelu, mida konkreetsetes testis kasutatakse. Juhul kui test kasutab ainult ühte ettemääratud DigiDoc faili, siis on selle sisu juba kopeeritud teststenaariumi sisse.

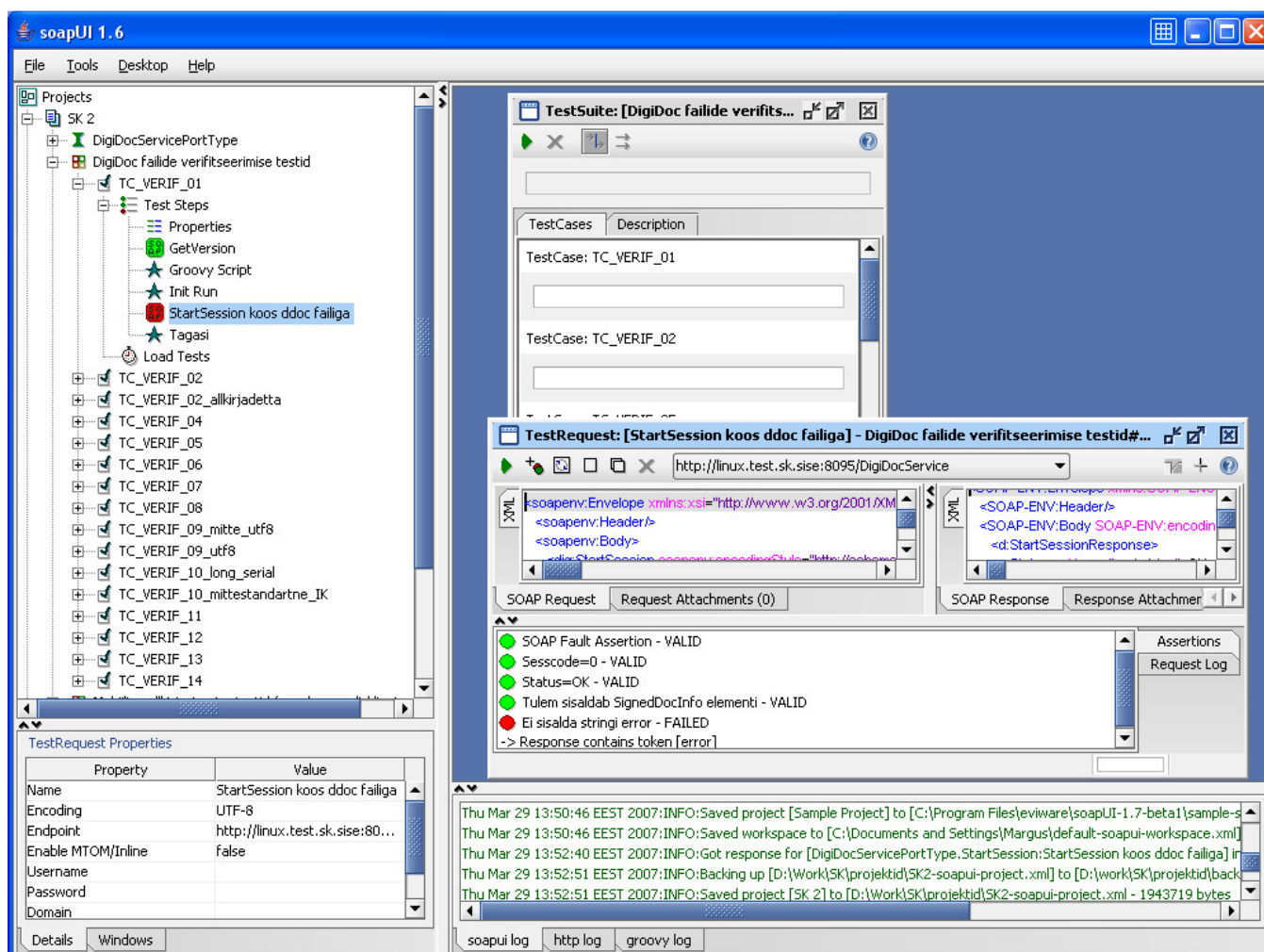
Mõned soapUI iseärasused

- testijuhtumi muutmisel kopeerides ühe testsammu ühest juhtumist teise juhtumisse ja taaskäivitamisel interaktiivselt võib juhtuda, et osad testi sammud kasutavad ühte teenuse lõpp-punkti ja osad teist lõpp-punkti.

DigiDocService automaattestid

Kasutusjuhend

Version 1.0 – 13.03.2009



Joonis 1 soapUI ekraanipilt koos DDS testidega

DigiDocService automaattestid

Kasutusjuhend

Versioon 1.0 – 13.03.2009



www.sk.ee

Muudatuste ajalugu

Kuupäev	Muutja	Muudatused
13.03.2009	Kristjan Jalukse	V1.0 - Faili aluseks on võetud 25.03.2007 Margus Härmi (ASA Quality Services OÜ) poolt valmistatud dokument versiooninumbriga 0.1.