

Automatize API com script atraente

Índice

[Introdução](#)

[Crie um projeto do soapUI](#)

[Crie um pedido do soapUI API](#)

[Crie um caso de teste do soapUI](#)

Introdução

Este documento descreve como criar programadores de aplicativo de um soapUI conecta o pedido (API) e como criar um caso de teste do soapUI que os laços sobre os passos de teste que automatizam o API pedem à série da política do quantum (QPS).

O caso de teste do soapUI do exemplo nos passos de teste deste implementa do artigo que leem um arquivo do subscritor ID e então criam e enviam um querySubscriberRequest a QPS.

Crie um projeto do soapUI

Antes que você comece este procedimento, instale o aplicativo do soapUI em seu desktop. Você pode transferir a instalação do soapUI executável de www.soapui.org.

Antes que você crie um pedido ou o caso de teste API você deve primeiramente criar um projeto do soapUI. Você precisa o arquivo do linguagem de descrição de serviços da web (WSDL) e o arquivo da descrição do esquema XML (XSD) a fim criar o projeto. WSDL especifica os API apoiados. Você pode normalmente obter WSDL e o XSD do QPS quando você executa estes comandos do equilibrador da carga (LB):

- `wget http://lbvip01:8080/ua/wsd/UnifiedApi.wsd`
- `wget http://lbvip01:8080/ua/wsd/UnifiedApi.xsd`

Armazene WSDL e o XSD no mesmo diretório no desktop onde você planeia executar o aplicativo do soapUI.

Termine estas etapas a fim criar o projeto do soapUI:

1. Escolha o **arquivo > projeto novo do soapUI** do indicador do soapUI:
2. No indicador novo do projeto do soapUI dê entrada com um nome para o projeto no campo de nome do projeto e entre no lugar onde o arquivo de WSDL é armazenado no campo inicial WSDL/WADL. Clique a **APROVAÇÃO** quando você é feito.

Crie um pedido do soapUI API

Termine estas etapas a fim criar um pedido do soapUI API:

1. Expanda o projeto que do soapUI você criou a fim ver os API. Você pode igualmente expandir um dos API a fim ver o pedido. Neste exemplo, o **mais querySubscriberRequest** é expandido:
2. Abra o pedido a fim ver o indicador do pedido com o XML que forma a pergunta. No indicador do pedido edite o endereço IP de Um ou Mais Servidores Cisco ICM NT de http:// ao endereço IP de Um ou Mais Servidores Cisco ICM NT e à porta. Esta é normalmente o endereço IP de Um ou Mais Servidores Cisco ICM NT lbvip01 e a porta onde você quer enviar o pedido enquanto este exemplo mostra:
3. Altere os campos no XML com os dados que você quer enviar em seu pedido. Neste exemplo, o pedido é um o mais querySubscriberRequest. Altere o subscritor ID para o subscritor que você quer perguntar e ajustar o showDetailedInformatin a **falso**:
4. Clique o **botão Run Button** verde na parte superior do indicador do pedido a fim executar a pergunta.

Crie um caso de teste do soapUI

Este procedimento explica como criar uma suite de teste que possa automatizar quando os API são enviados ao QPS.

Neste procedimento de exemplo a suite de teste dá laços sobre uma lista de subscritor ID e usa então aquele o subscritor ID no mais querySubscriberRequest que envie a QPS. A lista de subscritor ID é cada um em uma linha única em um arquivo de texto chamado **subid.txt**.

Termine estas etapas a fim criar a suite de teste:

1. No projeto que do soapUI você criou, cria uma suite de teste nova. Clicar com o botão direito no soapUI e escolha **TestSuite novo**.
2. Clicar com o botão direito a suite de teste e escolha o **TestCase novo**.
3. Clicar com o botão direito o caso de teste e escolha-o **adicionam a etapa > script atraente** a fim adicionar um passo de teste atraente do script. Nomeie-o **origem de dados**:
4. No arquivo da origem de dados cole este código. Este código lê o **C** do arquivo: **/subid.txt** **que contém um subscritor ID em cada linha:**

```
import com.eviware.soapui.support.XmlHolder def myTestCase = context.testCase
def counter,next,previous,sizeFile tickerEnumFile = new File("C:/subid.txt") //subscriber
IDs sepearted by new line (CR). List lines = tickerEnumFile.readlines() size =
lines.size.toInteger() propTestStep = myTestCase.getTestStepByName("Property - Looper")
// get the Property TestStep propTestStep.setPropertyValue("Total", size.toString())
counter = propTestStep.getPropertyValue("Count").toString() counter= counter.toInteger()
next = (counter > size-2? 0: counter+1) tempValue = lines[counter]
propTestStep.setPropertyValue("Value", tempValue) propTestStep.setPropertyValue
("Count", next.toString()) next++ log.info "Reading line : ${counter+1} /
$lines.size"propTestStep.setPropertyValue("Next", next.toString()) log.info
"Value '$tempValue' -- updated in $propTestStep.name" if (counter == size-1) {
propTestStep.setPropertyValue("StopLoop", "T") log.info "Setting the stoploop property
now..."}
else if (counter==0) { def runner = new
com.eviware.soapui.impl.wsdl.testcase.WsdlTestRunner
(testRunner.testCase, null) propTestStep.setPropertyValue("StopLoop", "F") } else{
propTestStep.setPropertyValue("StopLoop", "F") }
```

5. Clicar com o botão direito no passo de teste e escolha **adicionam a etapa > as propriedades**

- a fim adicionar um passo de teste da propriedade e nomeá-lo **propriedade - Looper**.
6. Adicionar estas propriedades definidas pelo utilizador do passo de teste do Looper:
TotalValor - Em nosso exemplo esta posse o subscritor ID lido do subscritor ID do arquivoContagemNext (Próximo)StopLoop
 7. Clicar com o botão direito no passo de teste e escolha **adicionam a etapa >** o **TestRequest** a fim adicionar um passo de teste do pedido do teste e escolher o pedido que você quer invocar:Neste exemplo, o mais querySubscriberRequest é usado.
 8. No pedido, o código da expansão substitui os valores de campo do que você pergunta.
Neste exemplo? **do SubscriberId=?** no mais querySubscriberRequest é substituído com o código **\$** da expansão **{propriedade - Looper#Value}**
(soap_test_req_expansion_code):**Propriedade - O Looper** é o nome da propriedade TestStep criado previamente e o **valor** mantém o assinante atual ID lido do arquivo do subscritor ID.
 9. Clicar com o botão direito no passo de teste e escolha **adicionam a etapa > script atraente** e nomeiam-na **laço dos dados**:
 10. Cole este código no **laço atraente dos dados** do script:


```
def myTestCase = context.testCase
def runner
propTestStep = myTestCase.getTestStepByName("Property - Looper")
endLoop = propTestStep.getPropertyValue("StopLoop").toString()
if (endLoop.toString() == "T" || endLoop.toString()=="True"
|| endLoop.toString()=="true")
{
log.info ("Exit Groovy Data Source Looper")
assert true
}
else
{
testRunner.gotoStepByName("Groovy Script - DataSource") //go to the DataSource
}

```
 11. Neste procedimento de exemplo, um atraso de 1000 Segundos entre cada laço é adicionado. Este passo é opcional.Com o atraso há agora cinco passos de teste:
 12. Clique o **botão Run Button** verde a fim executar os cinco passos de teste no indicador do TestCase.