Automatizar APIs com script Groovy

Contents

Introduction

Este documento descreve como criar uma solicitação soapUI Application Programmers Interface (API) e como criar um caso de teste de UI soap que faz loops sobre Etapas de teste que automatizam as solicitações de API para o Quantum Policy Suite (QPS).

O exemplo de caso de teste de interface de usuário soap neste artigo implementa etapas de teste que leem um arquivo de IDs de assinante e depois criam e enviam uma querySubscriberRequest para QPS.

Crie um projeto soapUI

Antes de iniciar este procedimento, instale o aplicativo soapUI em sua área de trabalho. Você pode fazer o download do arquivo executável de instalação da interface de usuário soap em www.soapui.org.

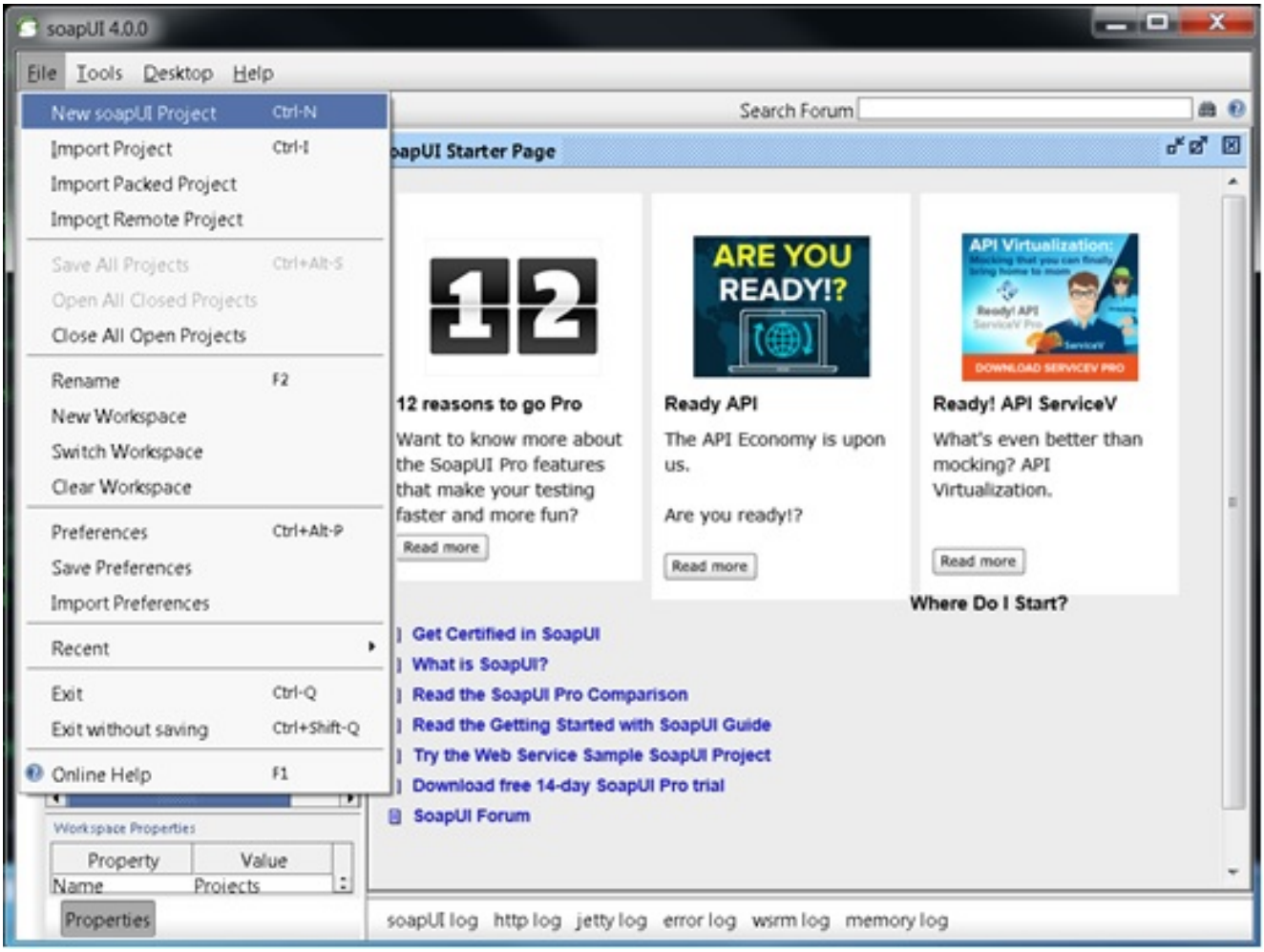
Antes de criar uma solicitação de API ou caso de teste, você deve primeiro criar um projeto soapUI. Você precisa do arquivo WSDL (Web Services Description Language) e do arquivo XSD (XML Schema Description) para criar o projeto. O WSDL especifica as APIs suportadas. Normalmente, você pode obter o WSDL e o XSD do QPS quando executa estes comandos a partir do Balanceador de carga (LB):

- **wget http://lbvip01:8080/ua/wsdl/UnifiedApi.wsdl**
- **wget http://lbvip01:8080/ua/wsdl/UnifiedApi.xsd**

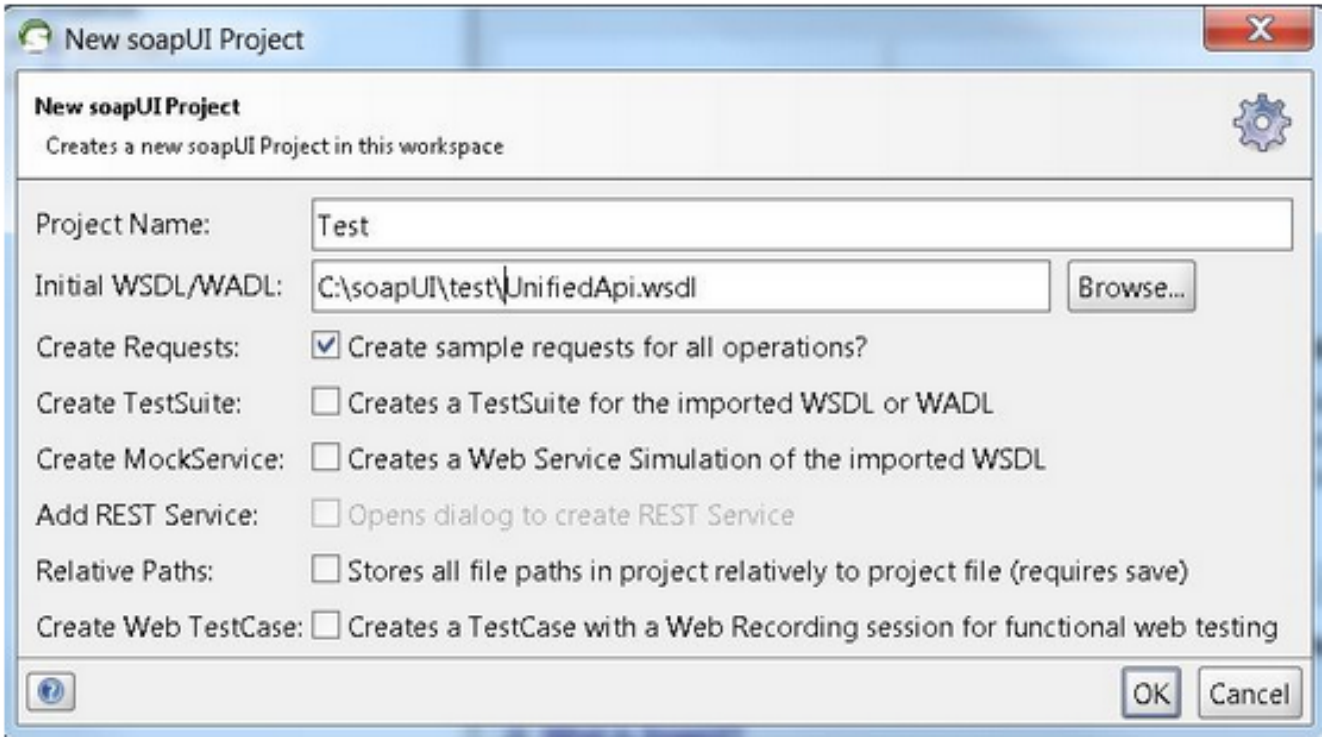
Armazene o WSDL e o XSD no mesmo diretório na área de trabalho onde pretende executar a aplicação soapUI.

Conclua estes passos para criar o projeto soapUI:

1. Escolha **File > New soapUI Project** na janela soapUI:



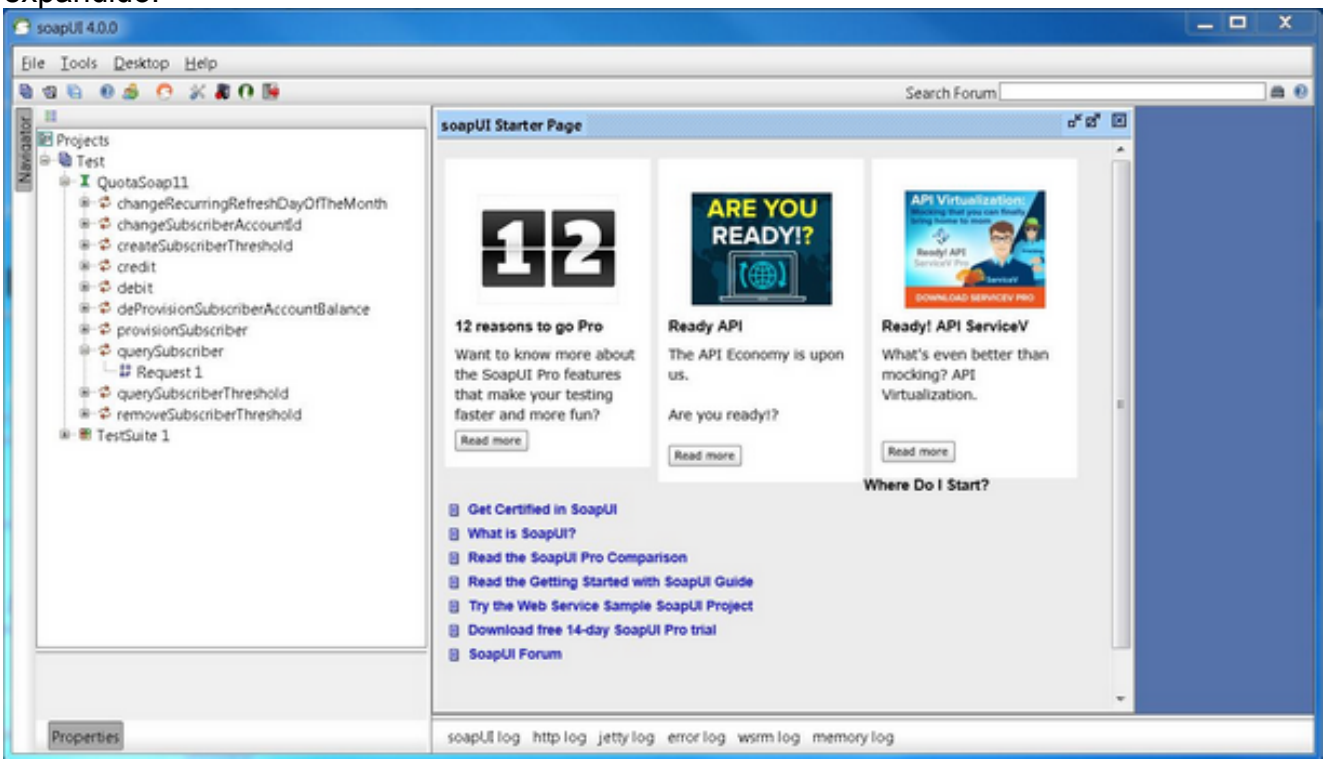
2. Na janela Novo projeto da IU de sabão, insira um nome para o projeto no campo Nome do projeto e insira o local onde o arquivo WSDL está armazenado no campo WSDL/WADL inicial. Clique em **OK** quando terminar.



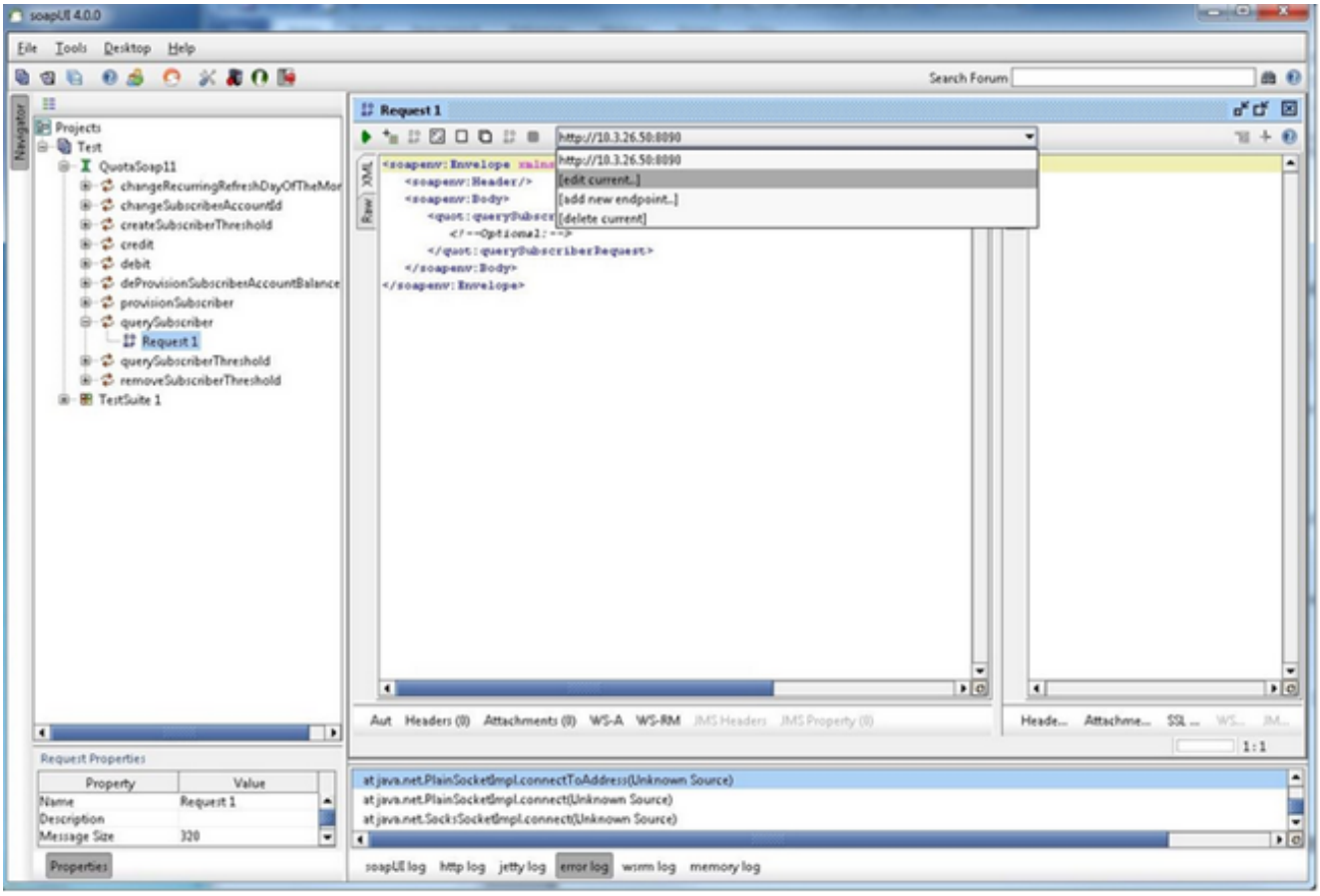
Crie uma solicitação de API soapUI

Conclua estes passos para criar uma solicitação de API soapUI:

1. Expanda o projeto soapUI que você criou para ver as APIs. Você também pode expandir uma das APIs para ver a solicitação.Neste exemplo, **querySubscriberRequest** é expandido:



2. Abra a solicitação para ver a janela de solicitação com o XML que forma a consulta. Na janela Solicitação, edite o endereço IP http:// para o endereço IP e a porta. Normalmente, esse é o endereço IP e a porta do lbvip01 em que você deseja enviar a solicitação, como mostrado neste
exemplo:



3. Modifique os campos no XML com os dados que você deseja enviar em sua solicitação. Neste exemplo, a solicitação é uma querySubscriberRequest. Modifique a ID do assinante que deseja consultar e defina showDetailedInformation como **false**:

4. Clique no botão **Executar** verde na parte superior da janela Solicitar para executar a consulta.

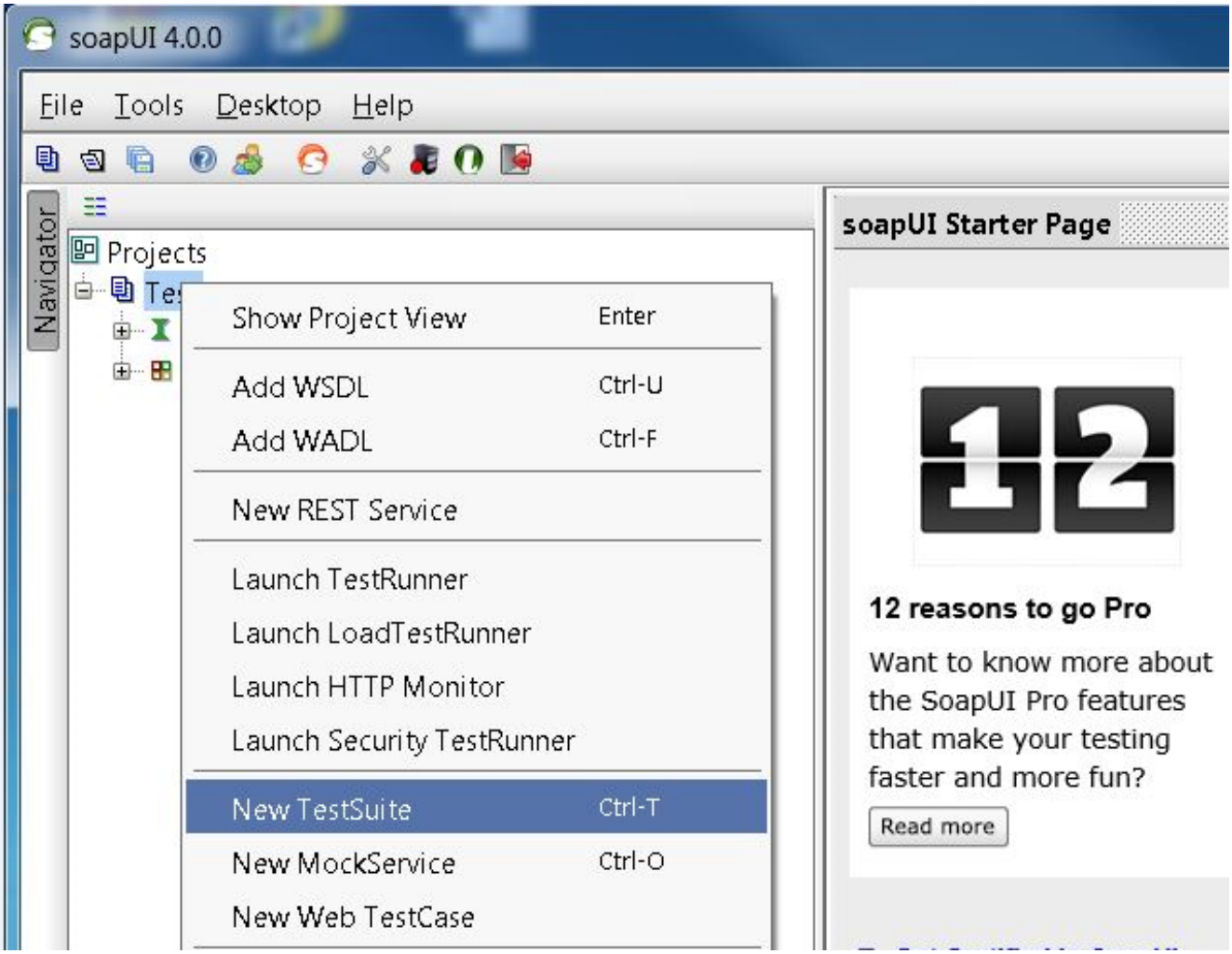
Criar um caso de teste de IU soap

Este procedimento explica como criar um conjunto de testes que pode automatizar quando as APIs são enviadas ao QPS.

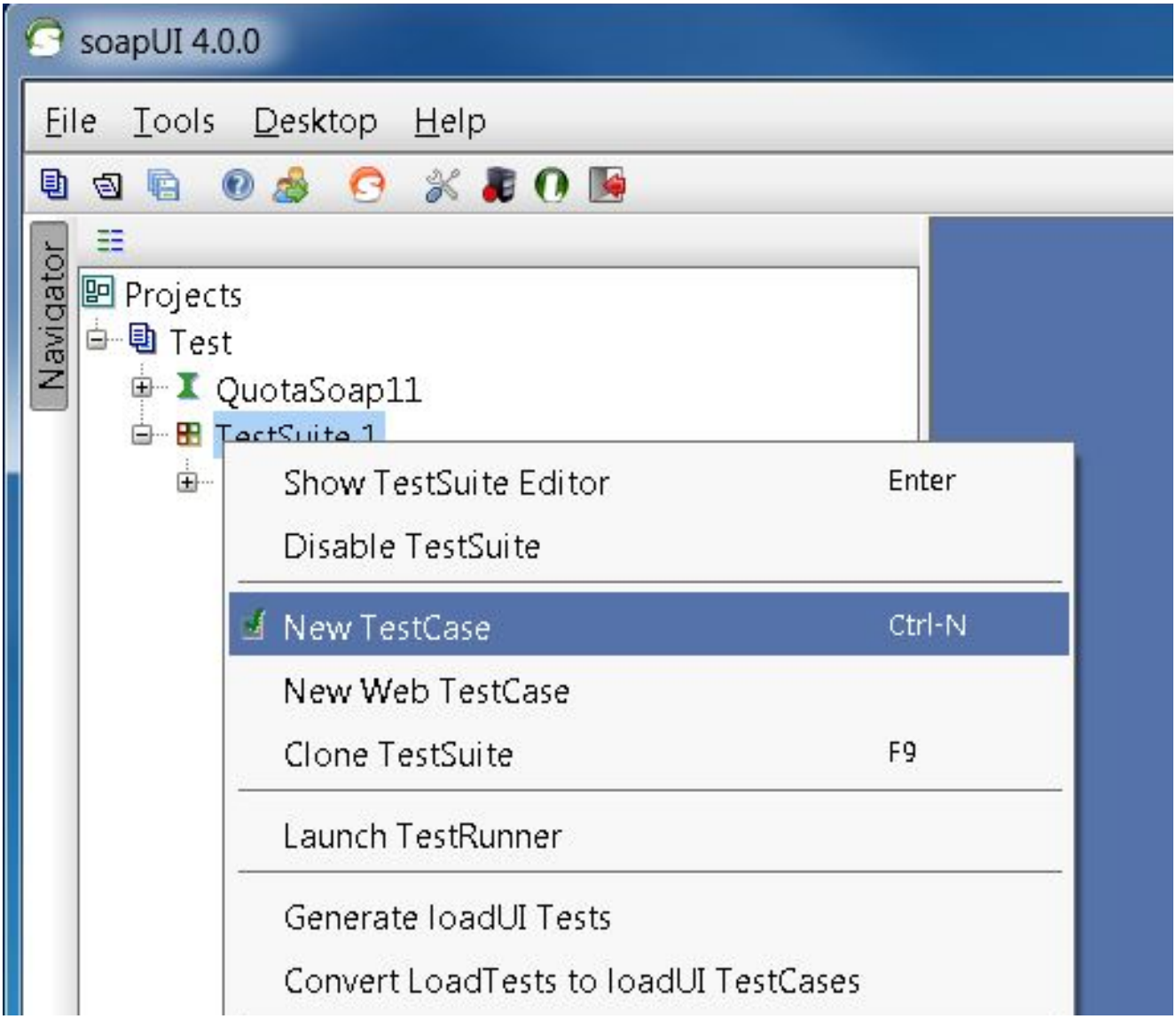
Neste procedimento de exemplo, o conjunto de testes faz loops em uma lista de IDs de assinante e, em seguida, usa essas IDs de assinante na querySubscriberRequest que ele envia ao QPS. A lista de IDs de assinante está em uma única linha em um arquivo de texto chamado **subid.txt**.

Conclua estes passos para criar o Test Suite:

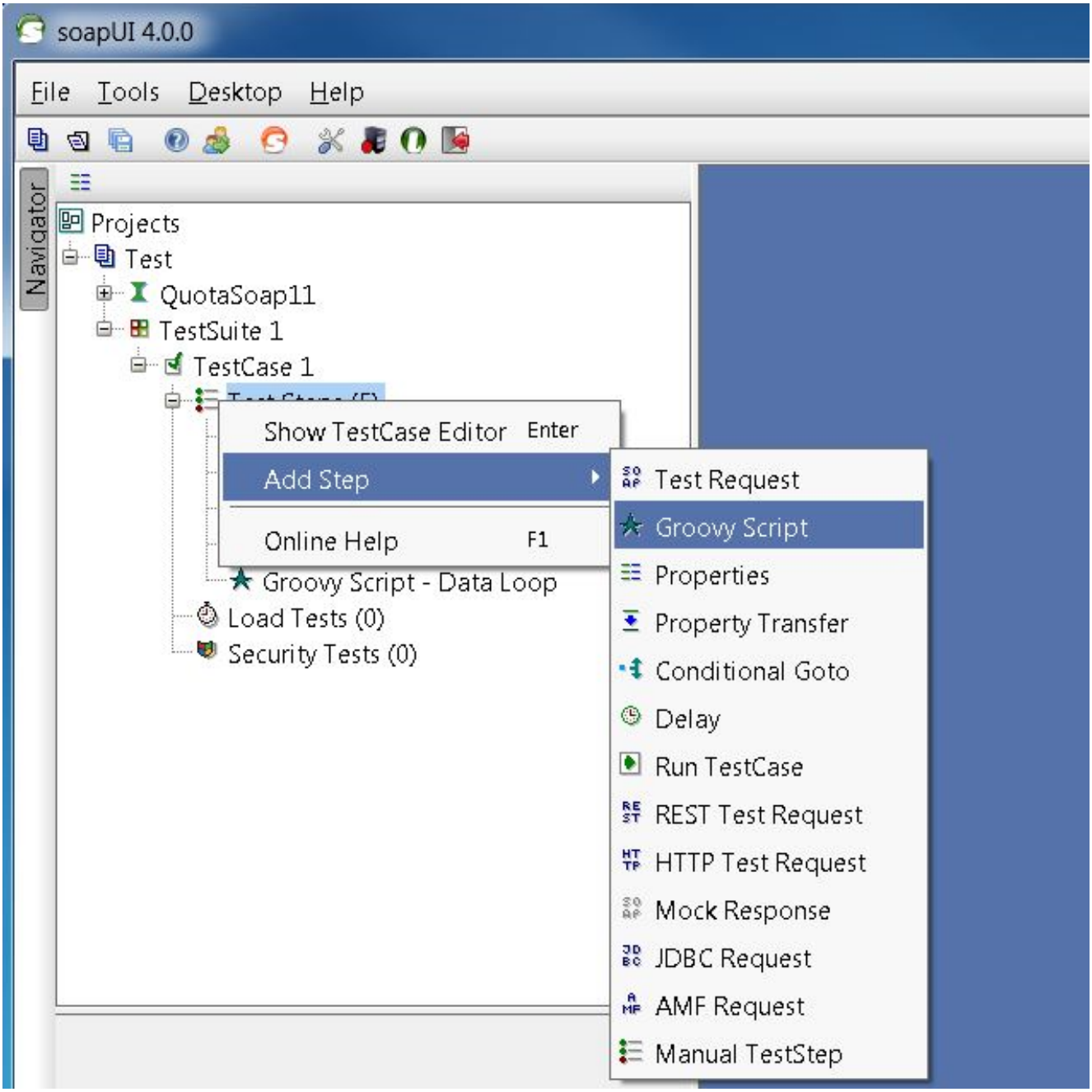
1. No projeto da IU da novela que você criou, crie um novo Test Suite. Clique com o botão direito do mouse na interface do usuário do sabão e escolha **New TestSuite**.



2. Clique com o botão direito do mouse no Test Suite e escolha **New TestCase**.



3. Clique com o botão direito do mouse no caso de teste e escolha **Add Step > Groovy Script** para adicionar uma etapa de teste do Groovy Script. Nomeie-o como **Fonte de Dados**:

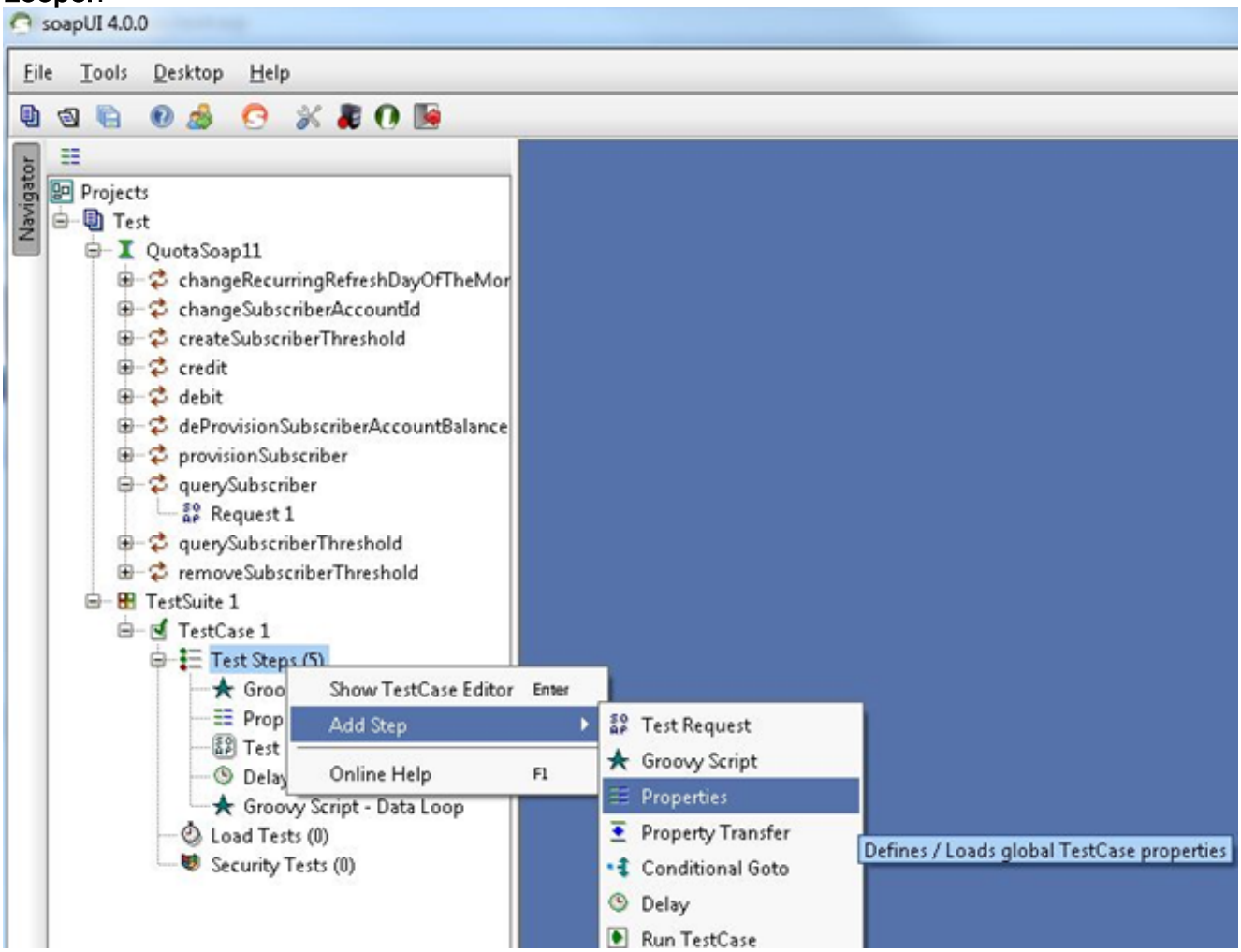


4. No arquivo Fonte de dados, cole este código. Este código lê o arquivo **C:/subid.txt** que contém um ID de assinante em cada linha:

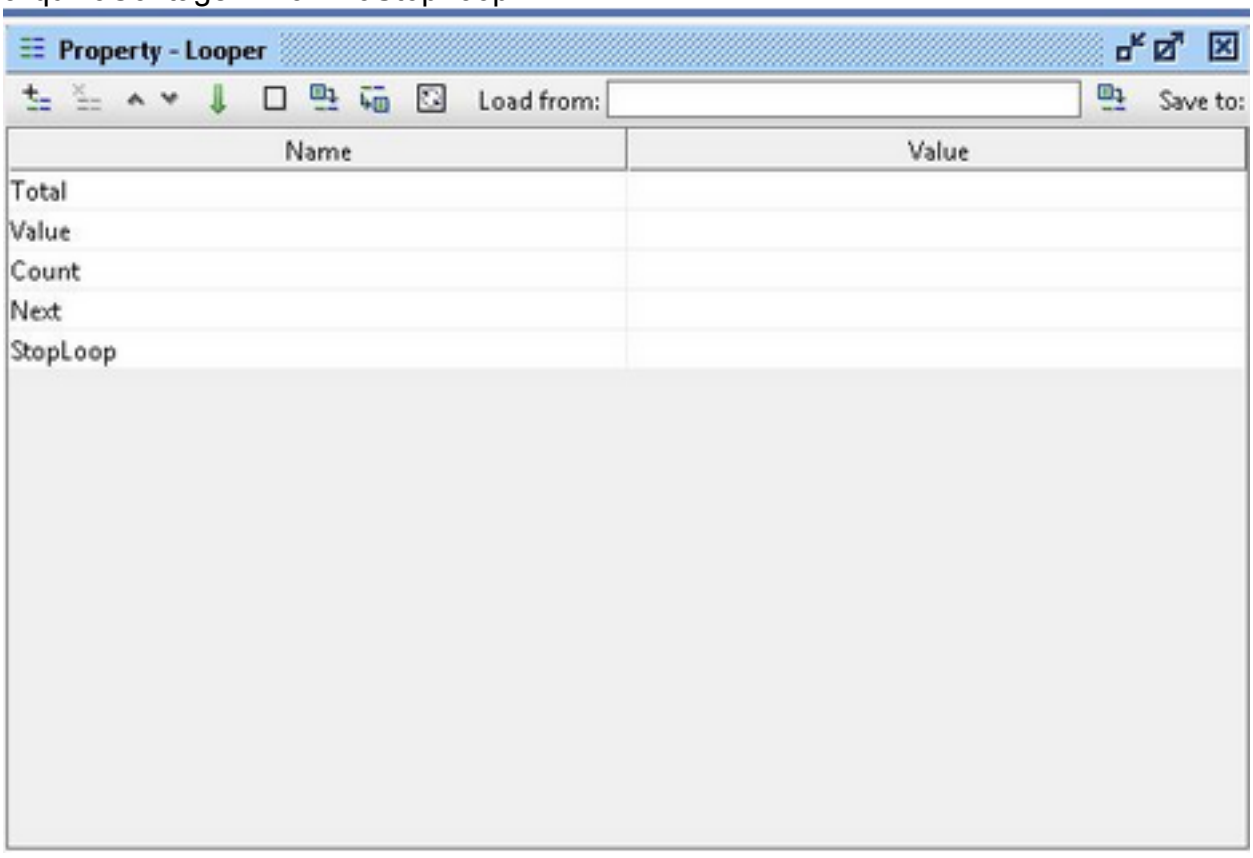
```
import com.eviware.soapui.support.XmlHolder def myTestCase = context.testCase
def counter,next,previous,sizeFile tickerEnumFile = new File("C:/subid.txt") //subscriber
IDs sepearted by new line (CR). List lines = tickerEnumFile.readLines() size =
lines.size.toInteger() propTestStep = myTestCase.getTestStepByName("Property - 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.WsdlTestCaseRunner
(testRunner.testCase, null) propTestStep.setPropertyValue("StopLoop", "F") } else{
propTestStep.setPropertyValue("StopLoop", "F") }
```

5. Clique com o botão direito do mouse na etapa de teste e escolha **Add Step > Properties** para adicionar uma etapa de teste de propriedade e nomeá-la como **Property -**

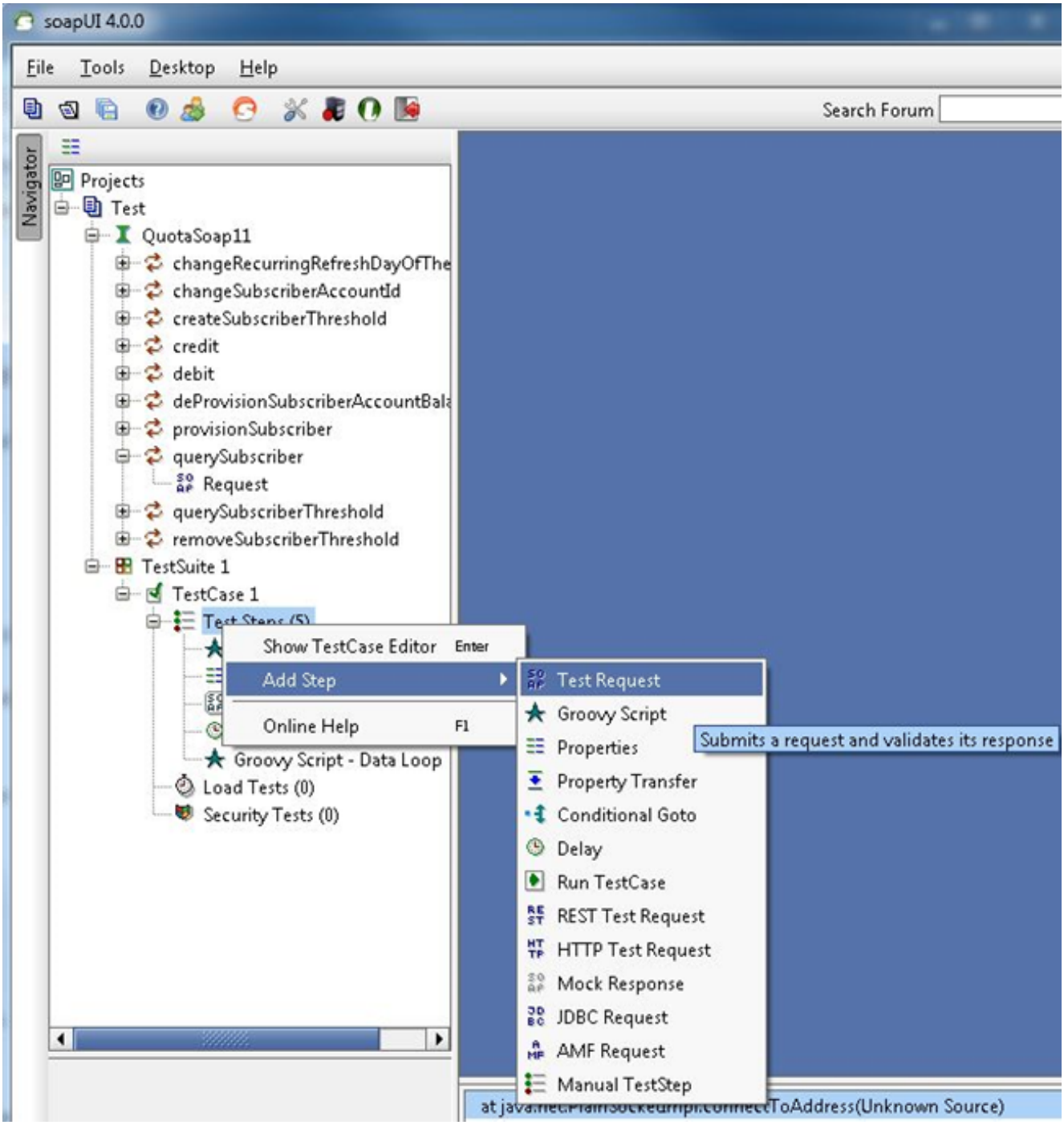
Looper.



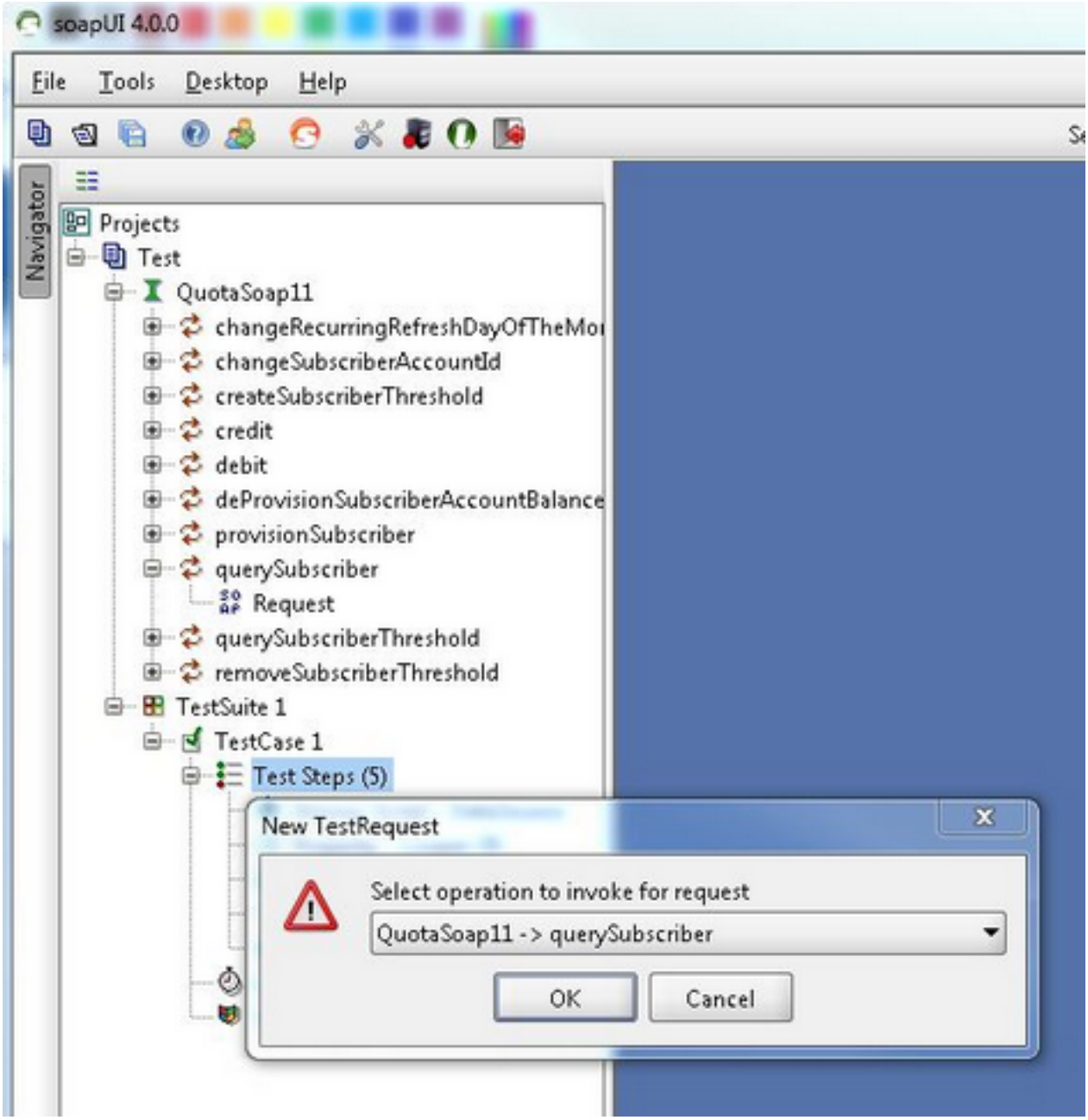
6. Adicione estas propriedades definidas pelo usuário da etapa de teste Looper: TotalValor - Em nosso exemplo, isso mantém a ID do assinante lida das IDs do assinante do arquivoContagemPróximoStopLoop



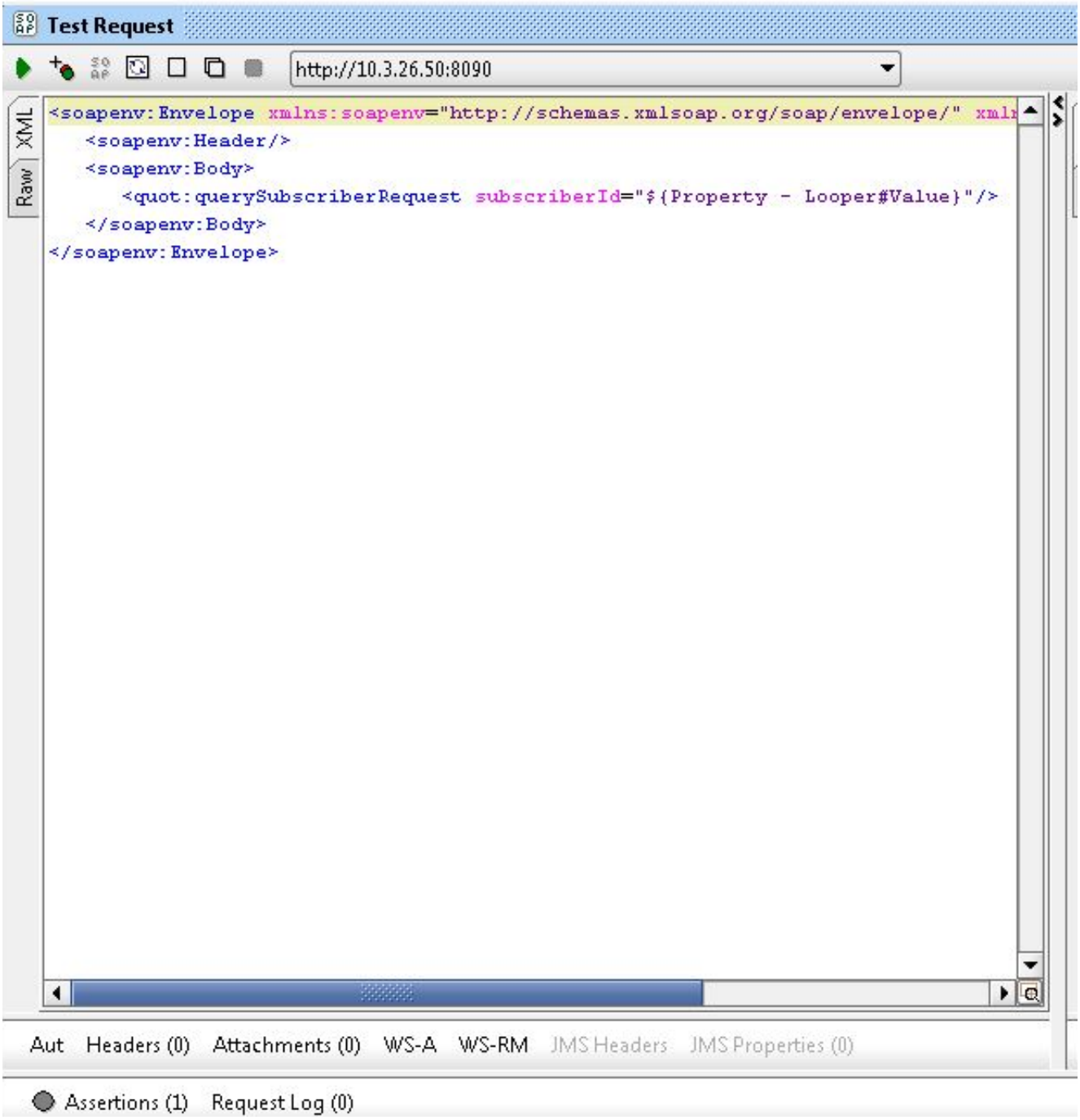
7. Clique com o botão direito do mouse na etapa de teste e escolha **Add Step > TestRequest** para adicionar uma etapa de teste de solicitação e escolha a solicitação que deseja invocar:



Neste exemplo, querySubscriberRequest é usado.

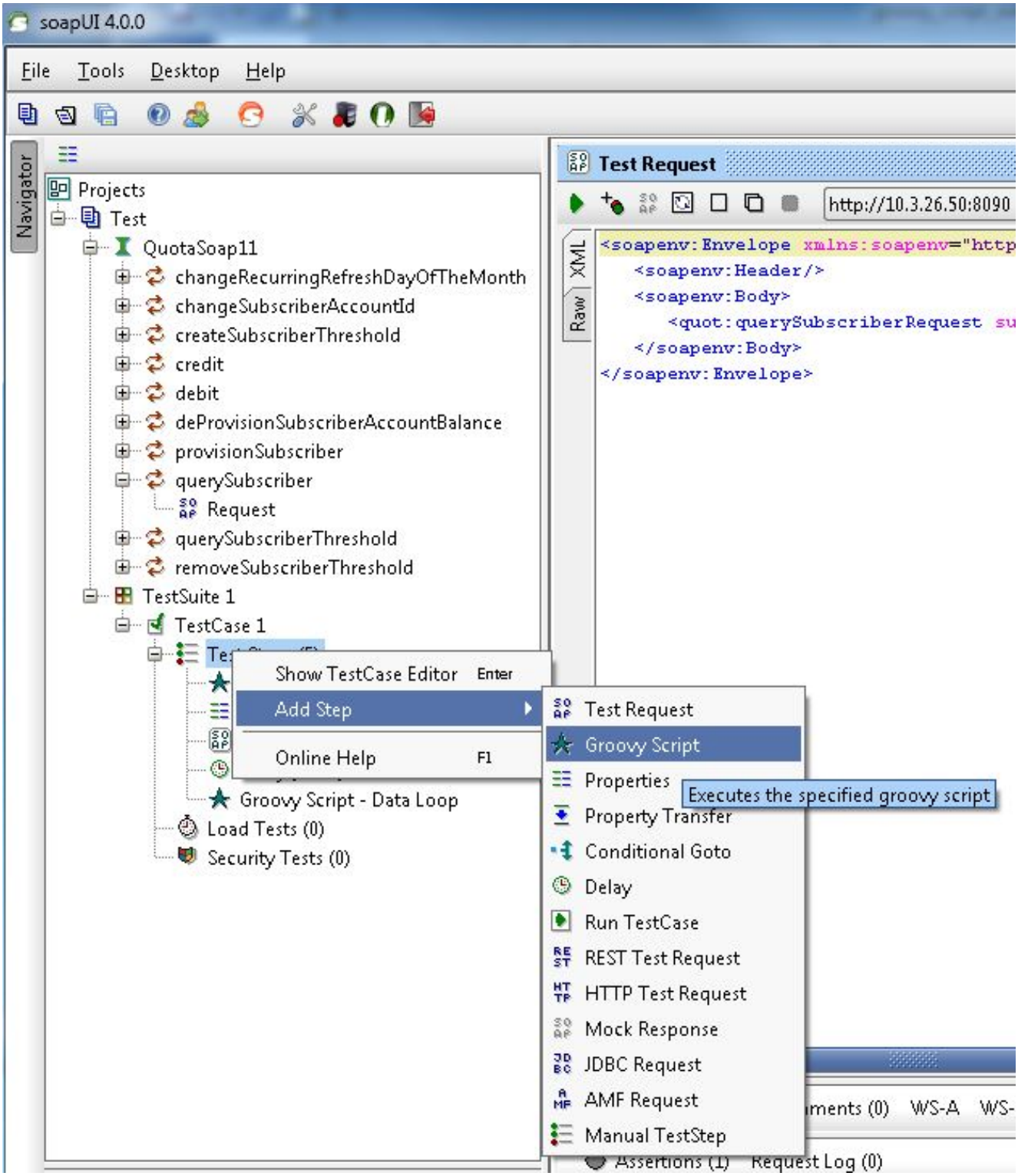


8. Na solicitação, o código de expansão substitui os valores dos campos do que você consulta. Neste exemplo, o **?** do **SubscriberId=?** na querySubscriberRequest é substituído pelo código de expansão **${Property - Looper#Value}**
(soap_test_req_expand_code):



Propriedade - Looper é o nome da Propriedade TestStep criada anteriormente e **Value** mantém a ID do assinante atual lida do arquivo de IDs do assinante.

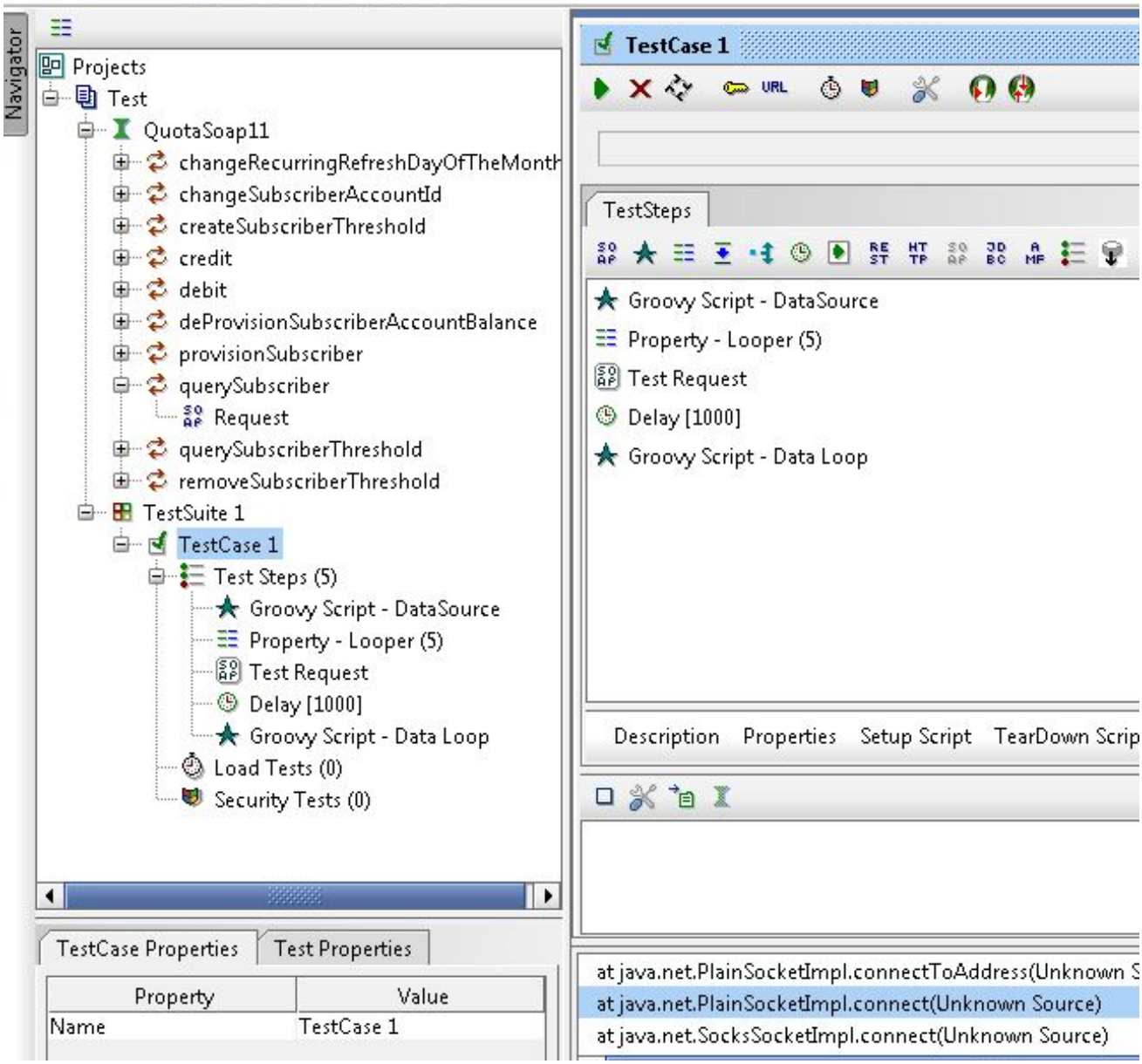
9. Clique com o botão direito do mouse na etapa de teste e escolha **Add Step > Groovy Script** e nomeie-o como **Data Loop**:



10. Cole este código no Groovy Script Data Loop:

```
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 ms entre cada loop é adicionado. This step is optional.Com o atraso, há cinco etapas de teste:



12. Clique no botão **Executar** verde para executar as cinco Etapas de teste na janela TestCase.