

Automatische API's met Groovy Script

Inhoud

[Inleiding](#)

[Een soapUI-project maken](#)

[Een API-aanvraag maken voor een sessie](#)

[Een test-case met een zeepUI maken](#)

Inleiding

Dit document beschrijft hoe u een API-verzoek (soapUI) van toepassingsprogrammeurs kunt maken en hoe u een API-test (soapUI-testcase) kunt maken die via Test-stappen loopt om de API-verzoeken om Quantum Policy Suite (QPS) te automatiseren.

De voorbeeldcase van de test in dit artikel implementeert een Test Stappen die een bestand van abonnee ID's lezen en dan een querySubscriberRequest naar QPS maken en verzenden.

Een soapUI-project maken

Voordat u met deze procedure begint, installeert u de soapUI-toepassing op uw bureaublad. U kunt de installatie soapUI van www.soapui.org downloaden.

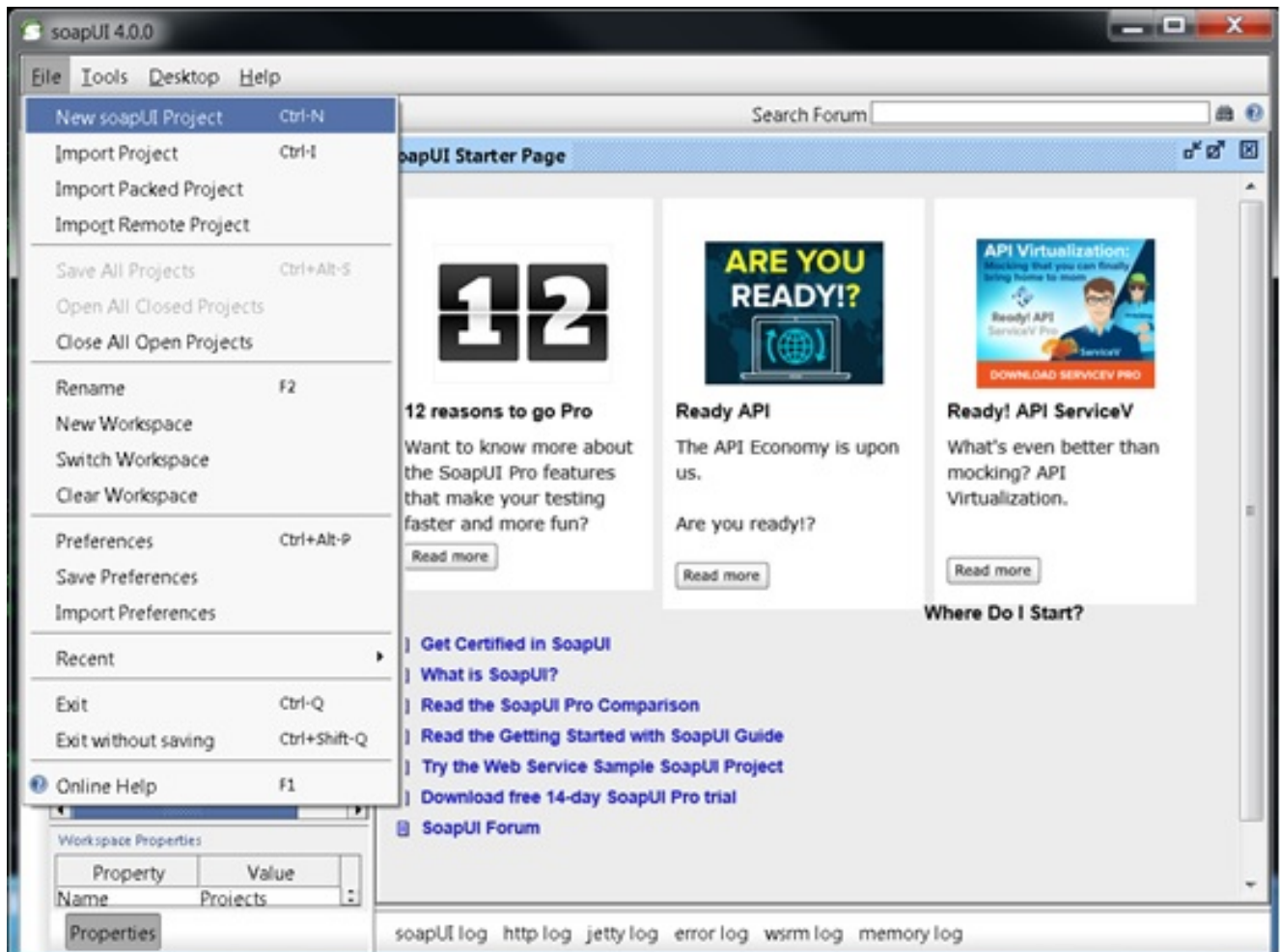
Voordat u een API-aanvraag of een Test-case maakt, moet u eerst een soapUI-project maken. U hebt het WSDL-bestand (Web Services Description Language) en het XSD-bestand (XML Schema Description) nodig om het project te maken. De WSDL specificeert de ondersteunde API's. U kunt de WSDL en XSD doorgaans van het QPS verkrijgen wanneer u deze opdrachten vanuit de taakbalk (LB) voert:

- <http://lbvip01:8080/ua/wsd/UnifiedApi.wsd>
- <http://lbvip01:8080/ua/wsd/UnifiedApi.xsd>

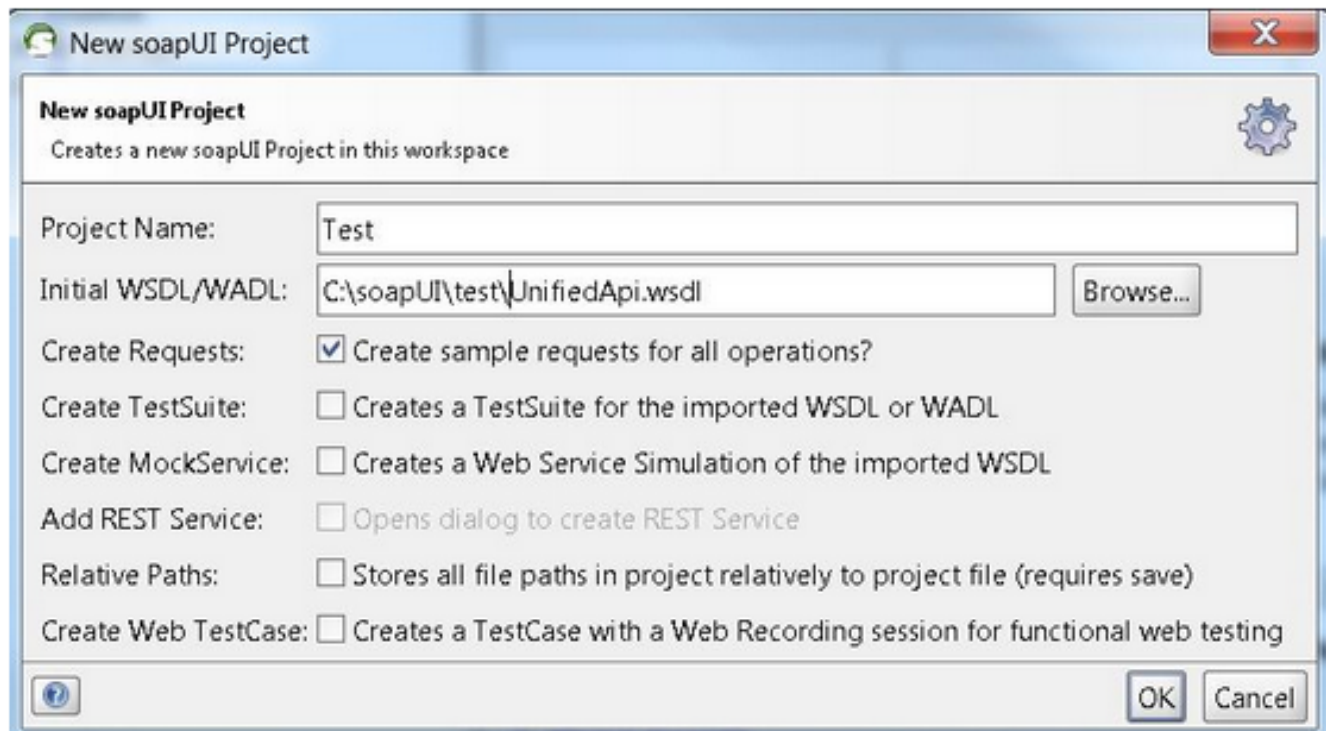
Bewaar de WSDL en XSD in de zelfde folder op het desktop waar u van plan bent om de toepassing soapUI te leiden.

Voltooi deze stappen om het soapUI-project te maken:

1. **Bestand kiezen > Nieuw soapUI-project** uit het venster soapUI:



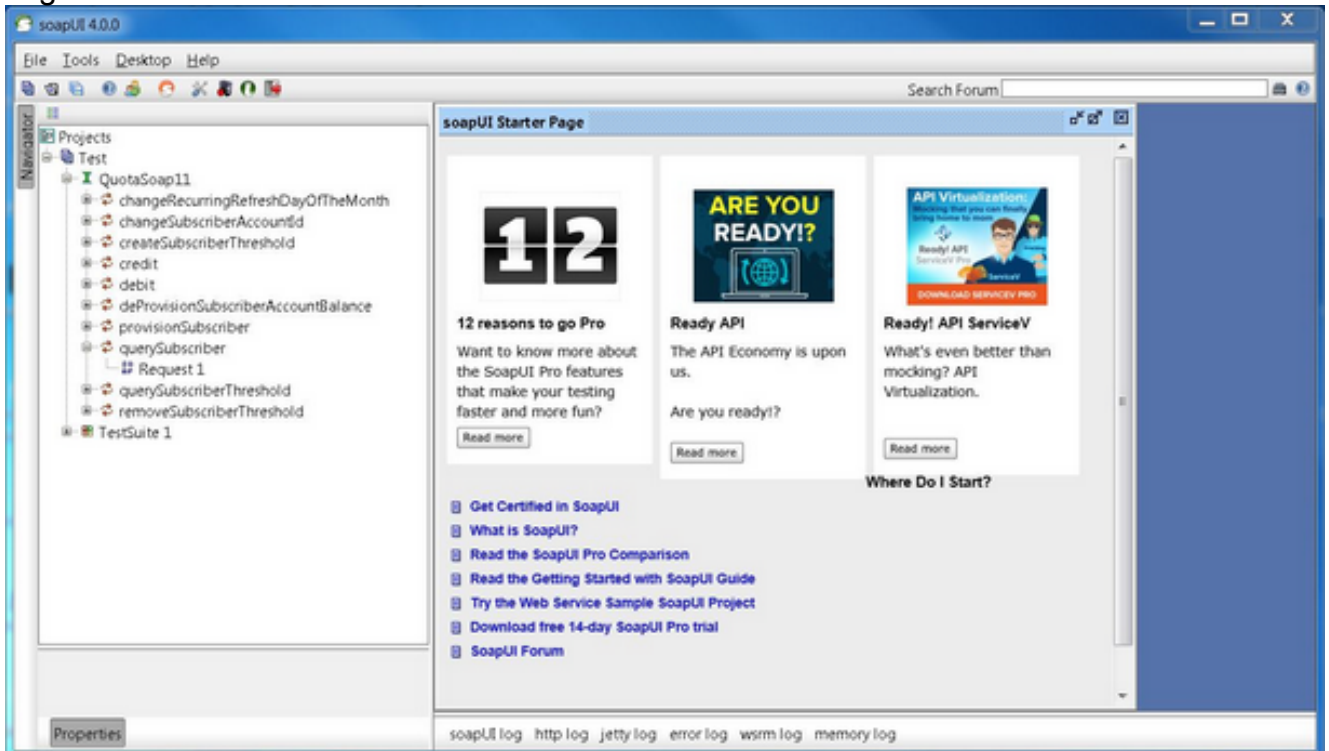
2. In het venster New soapUI Project typt u een naam voor het project in het veld Naam van het project en voert u de locatie in waar het WSDL-bestand is opgeslagen in het veld Initiële WSDL/WADL. Klik op OK wanneer u klaar bent.



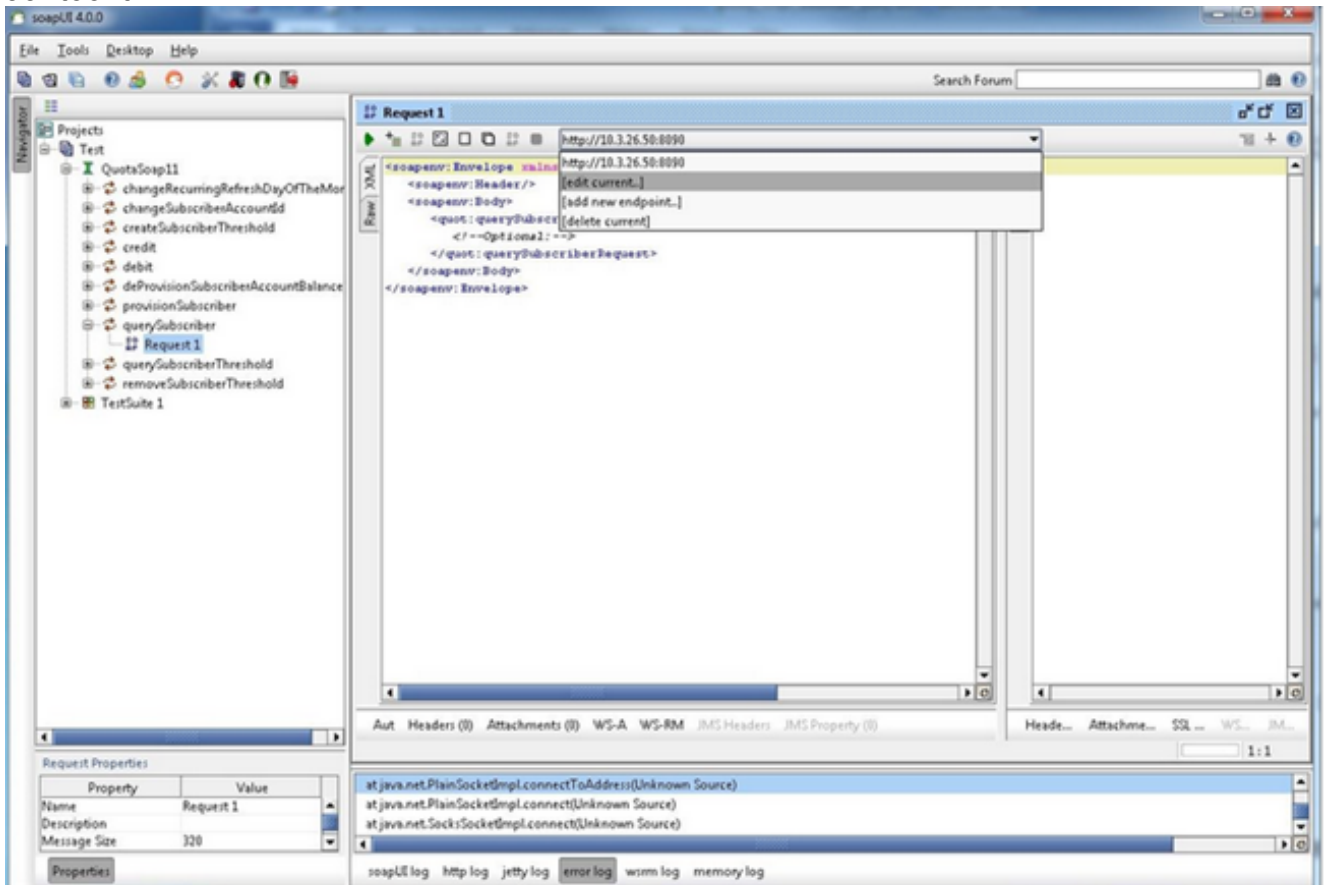
Een API-aanvraag maken voor een sessie

Voltooi deze stappen om een aanvraag voor een soapUI-applicatie te maken:

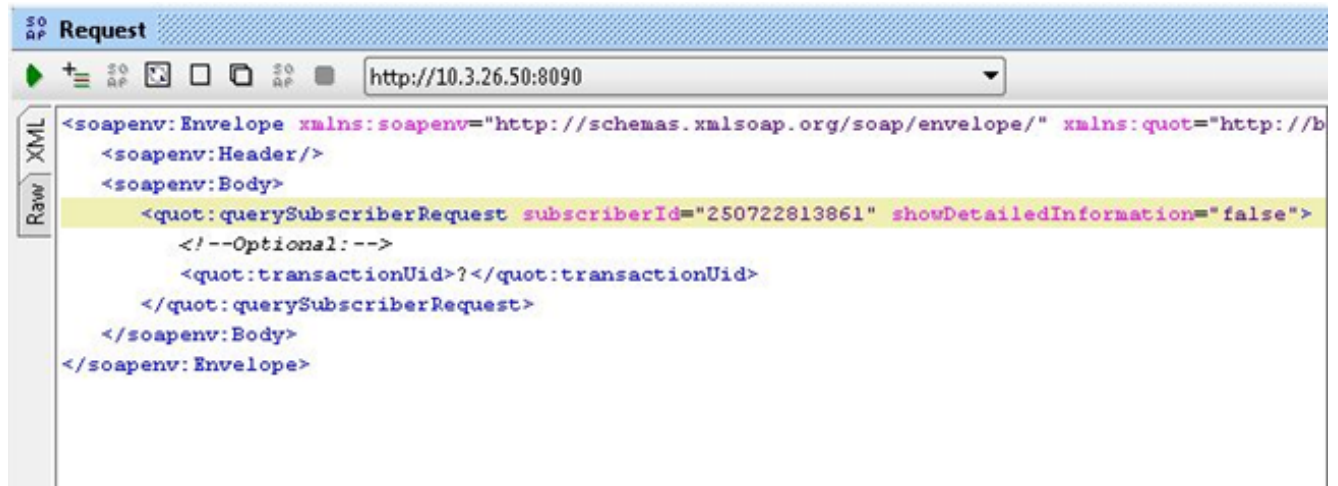
1. Uitbreid het project soapUI dat u hebt gemaakt om de APIs te zien. U kunt ook één van de API's uitbreiden om het verzoek te zien. In dit voorbeeld wordt `querySubscriberApplication` uitgebreid:



2. Open het verzoek om het verzoekvenster met XML te zien dat de query vormt. Bewerk in het venster Aanvraag het `http://` IP-adres naar het IP-adres en de poort. Dit is normaal het IP-adres en de IP-poort van `lbvip01` waar u het verzoek wilt verzenden zoals dit voorbeeld aantoont:



3. Wijzig de velden in XML met de gegevens die u in uw verzoek wilt verzenden. In dit voorbeeld is het verzoek een querySubscriberApplication. Wijzig de abonnee-ID voor de abonnee die u wilt vragen en stel deze in met gedetailleerde informatie op fout:



4. Klik op de knop groen uitvoeren boven in het venster Aanvragen om de query uit te voeren.

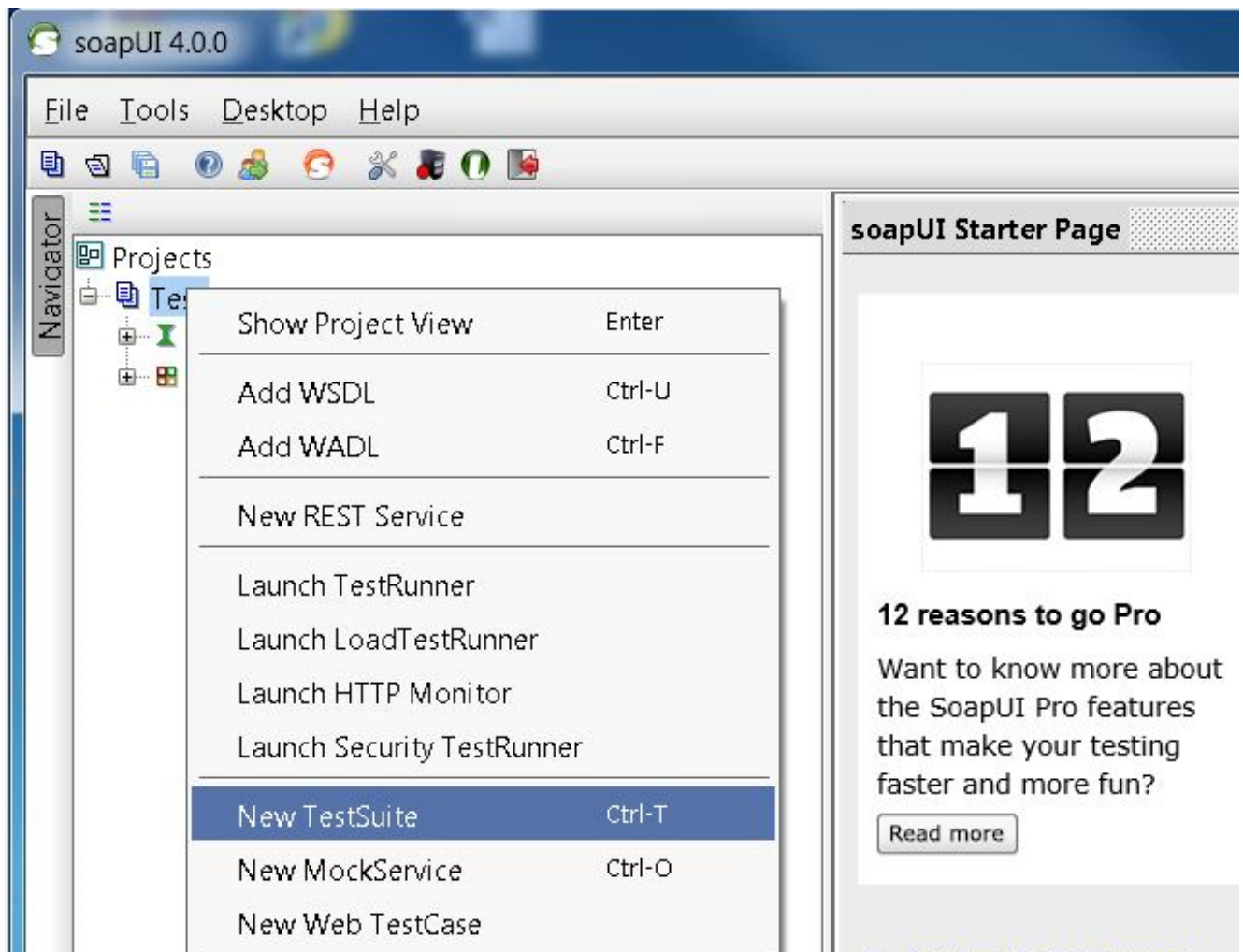
Een test-case met een zeepUI maken

Deze procedure legt uit hoe een testreeks te maken die kan automatiseren wanneer APIs naar de QPS worden verzonden.

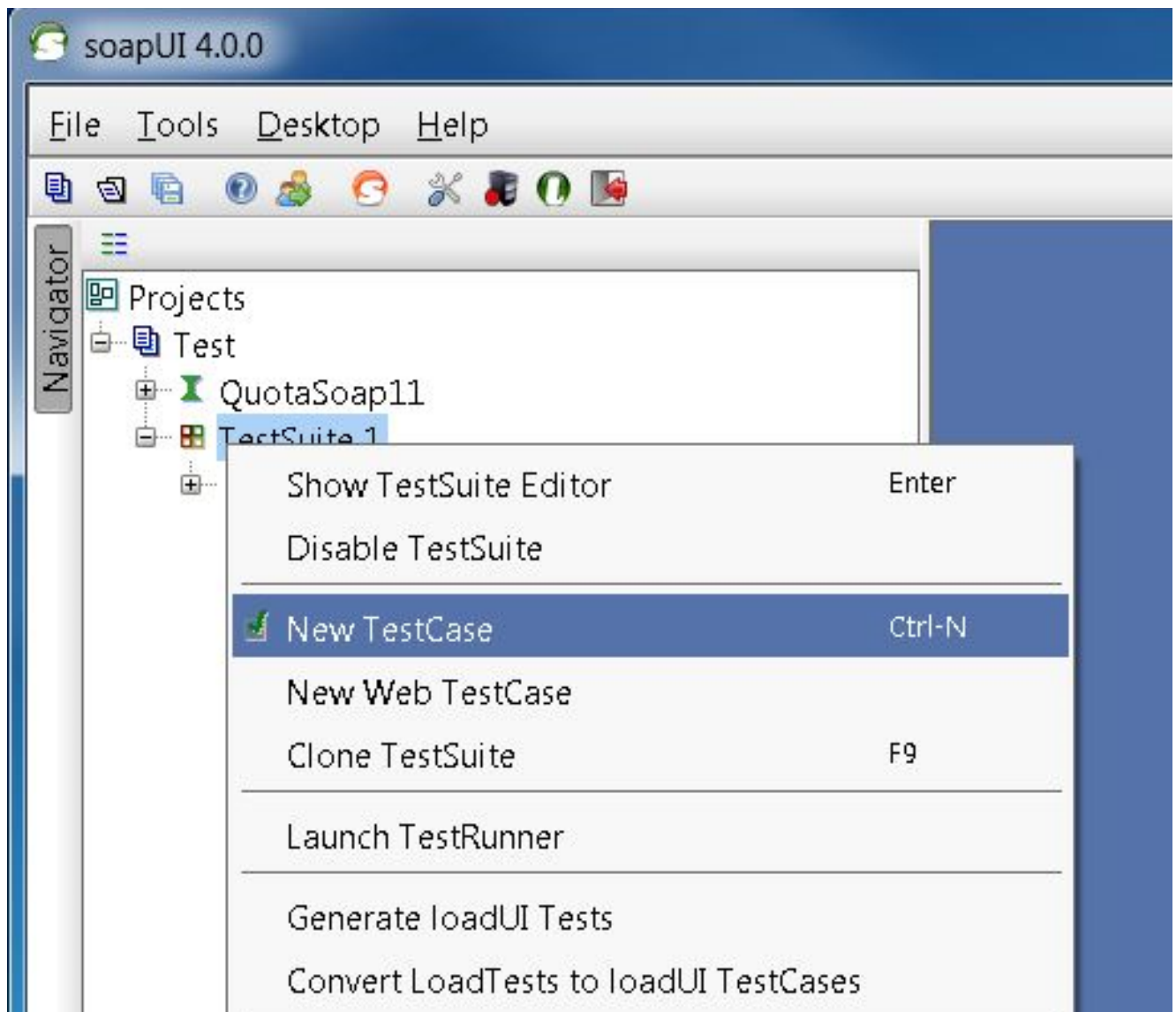
In deze voorbeeldprocedure loopt de testsuite over een lijst van abonnee-id's en gebruikt deze abonnee-id's vervolgens in de querySubscriberApplication en wordt deze naar QPS verzonden. De lijst van abonnee-ID's staat elk op één regel in een tekstbestand dat **subid.txt** wordt genoemd.

Voltooi deze stappen om de Test Suite te maken:

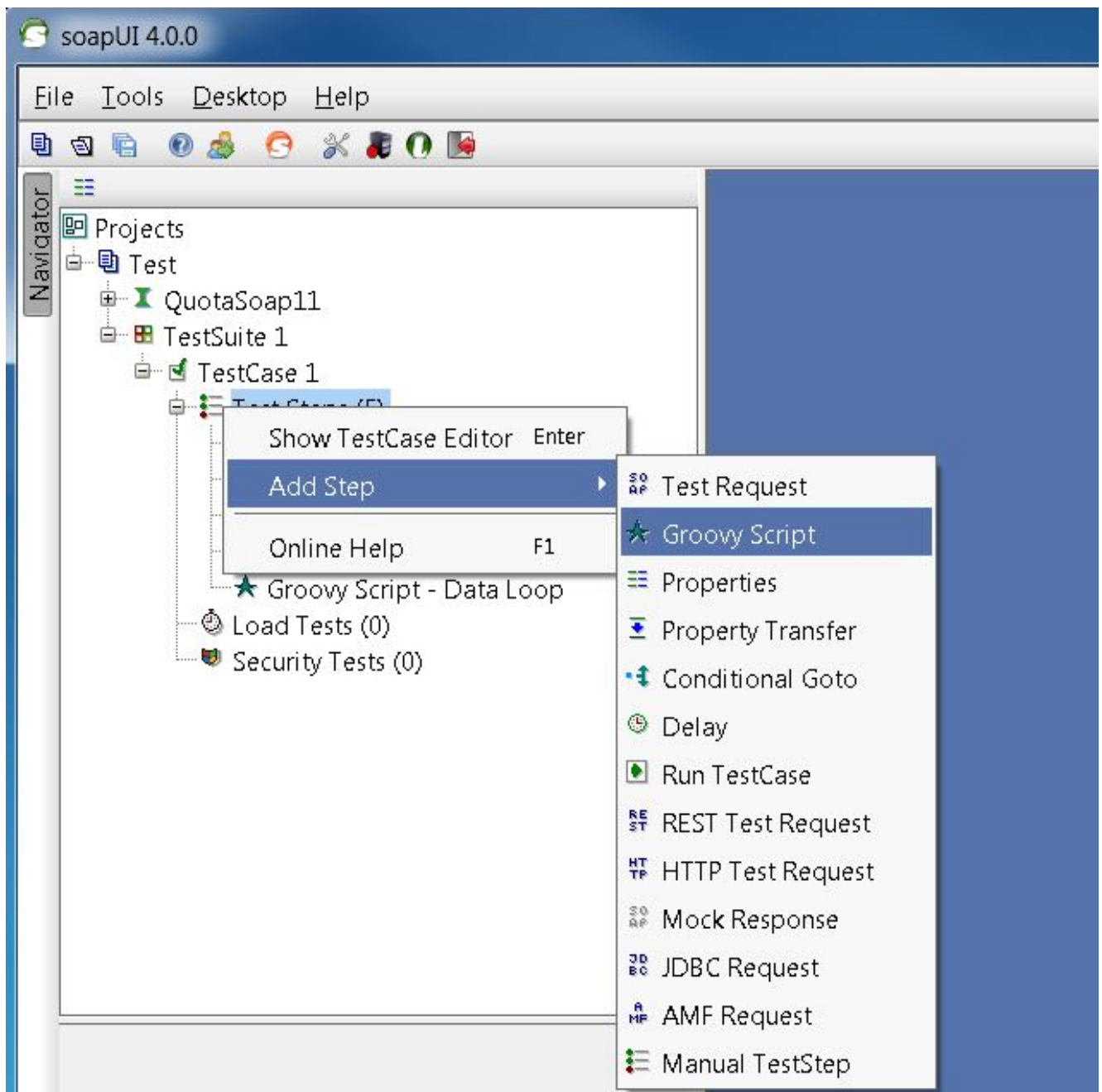
1. In het soapUI-project dat u hebt gemaakt, creëert u een nieuwe Test Suite. Klik met de rechtermuisknop op de soapUI en kies **New TestSuite**.



2. Klik met de rechtermuisknop op de Test Suite en kies **New TestCase**.



3. Klik met de rechtermuisknop op de Test Case en kies **Step > Groovy Script** toevoegen om een proefstap voor Groovy Script toe te voegen. Naam it **Data Source**:

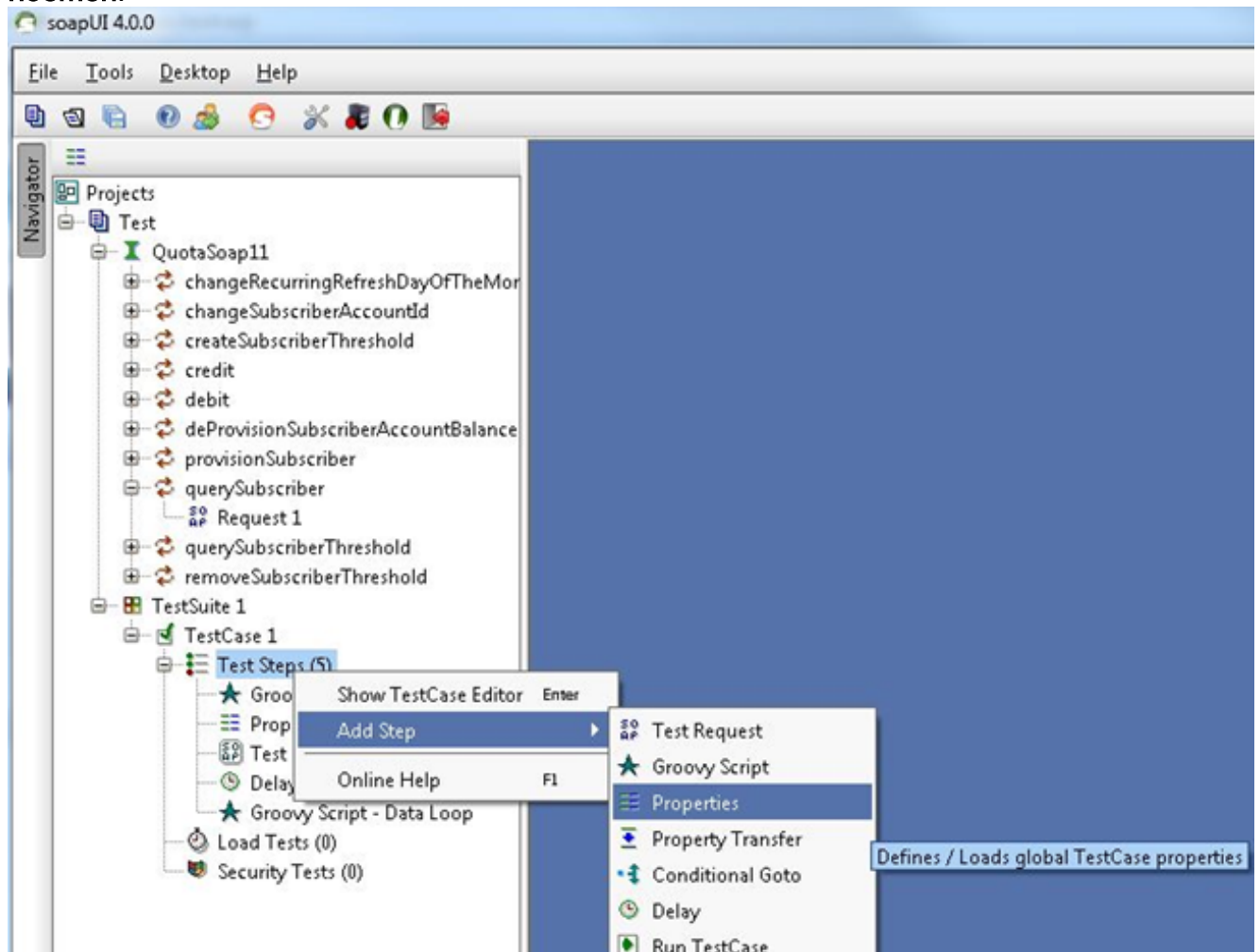


4. In het Data Source file paste deze code aan. Deze code leest bestand **C:/subid.txt** dat op elke regel een abonnee-ID bevat:

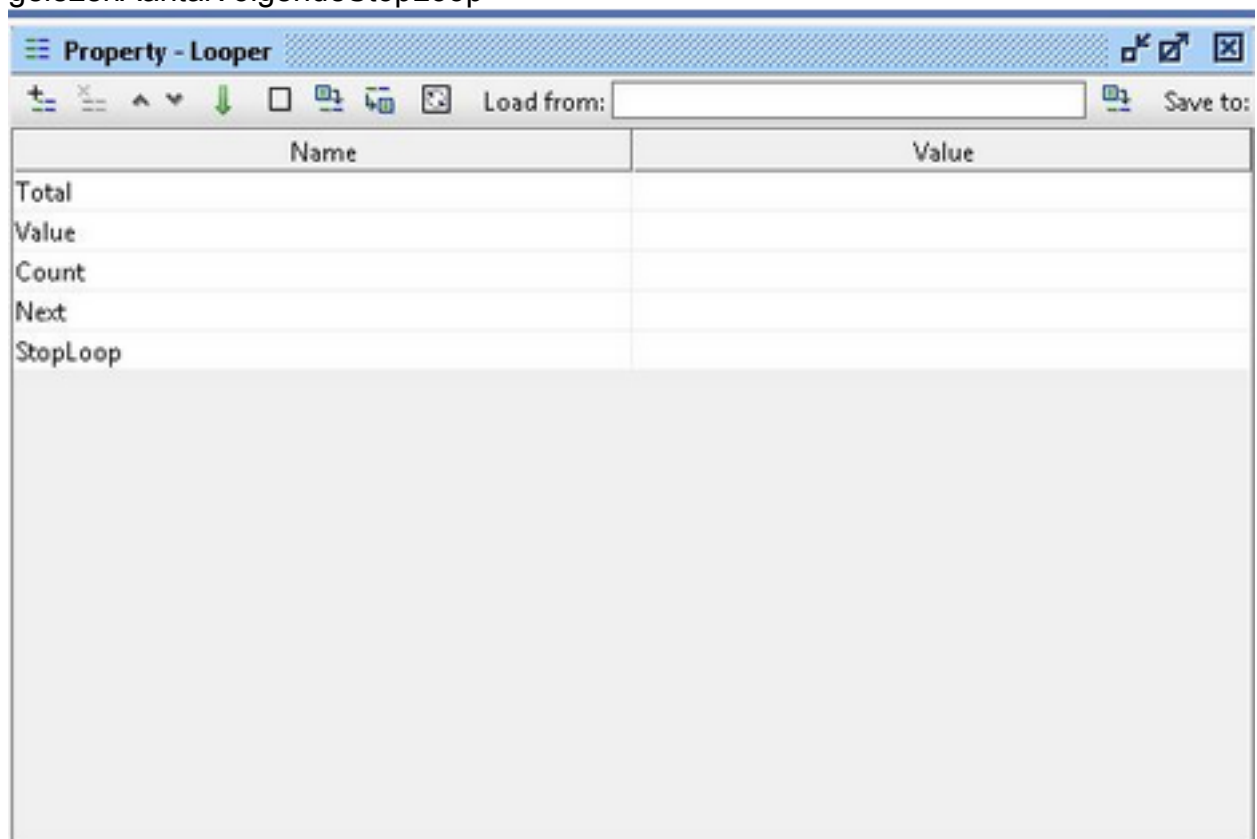
```
import com.eviware.soapui.support.XmlHolder def myTestCase = context.testCase
def counter,next,previous,sizeFile tickerEnumFile = new File("C:/subid.txt") //subscriber
IDs separted by new line (CR). List lines = tickerEnumFile.readlines() size =
lines.size.toInteger() propTestStep = myTestCase.getTestStepByName("Property - 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. Klik met de rechtermuisknop op de Test Stap en kies **Stap > Eigenschappen toevoegen** om een teststap van het Eigenschap toe te voegen en het **Eigenschap - Looper** te

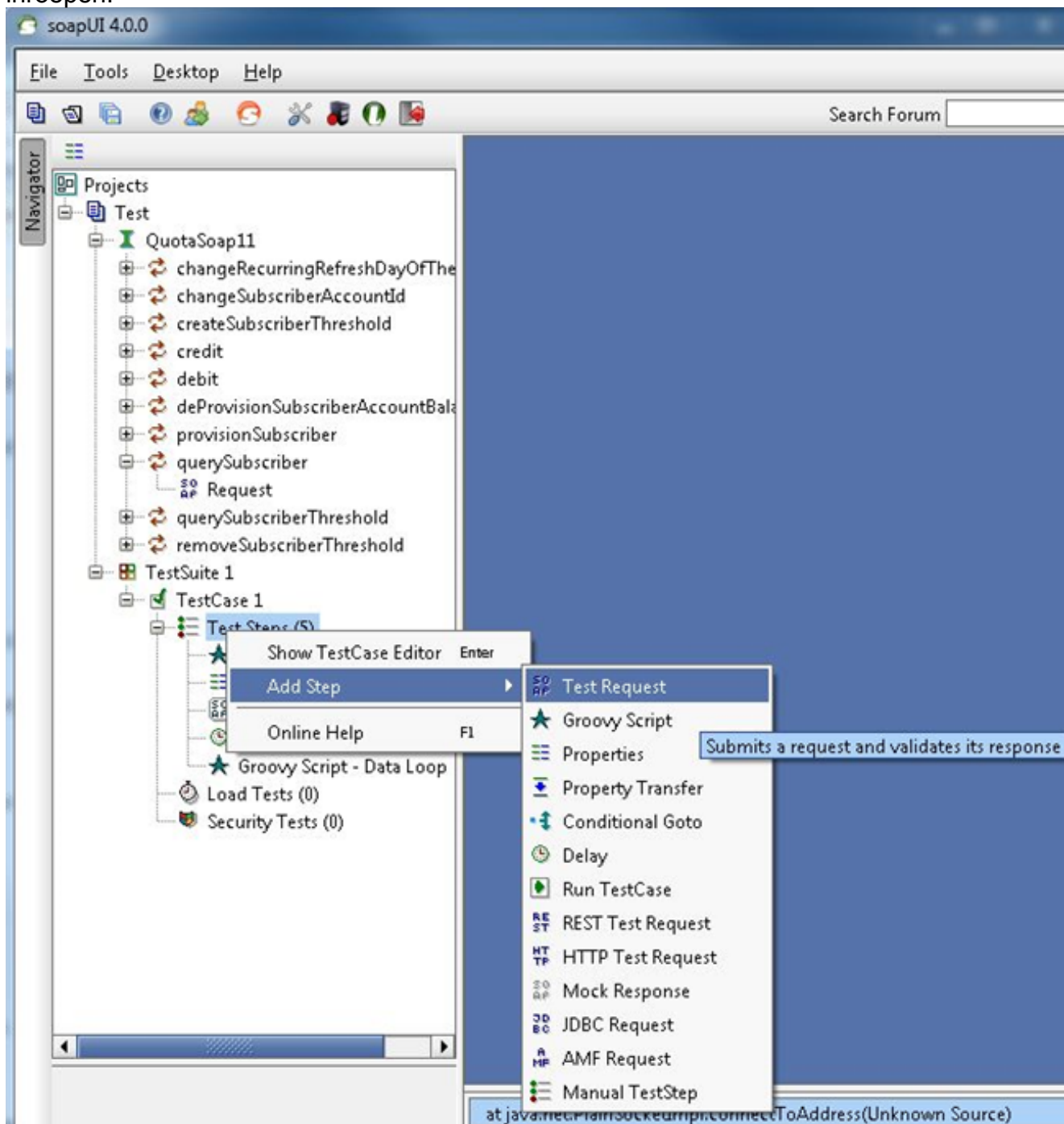
noemen.



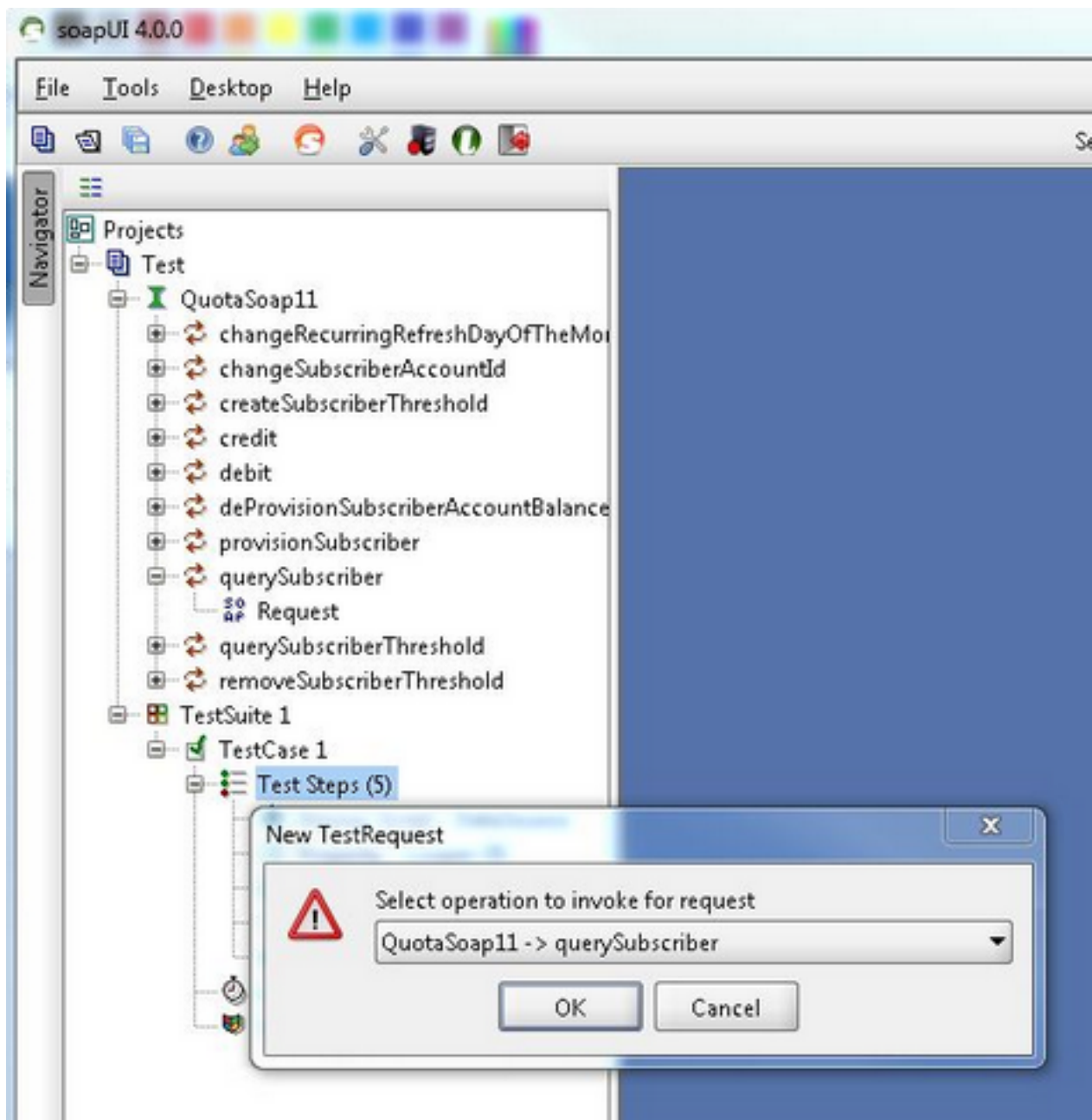
6. Voeg deze gebruiker bepaalde eigenschappen van de Looper teststap toe: TotaalWaarde - In ons voorbeeld is dit het nummer van de abonnee-ID die van de bestand-abonnee-ID's is gelezen
AantalVolgendeStopLoop



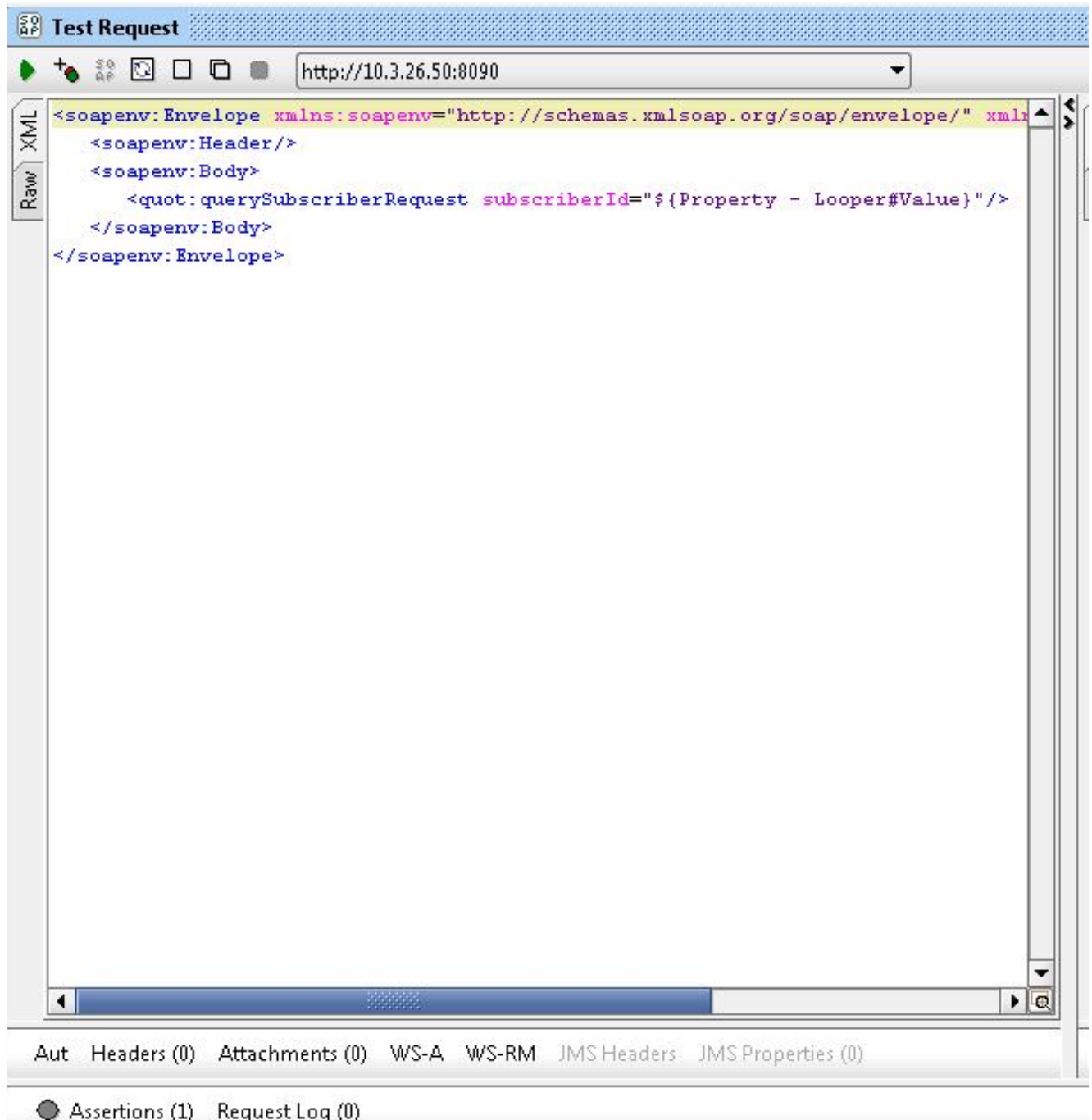
7. Klik met de rechtermuisknop op de Test Step en kies **Add Step > TestApplication** om een teststep voor testaanvraag toe te voegen en kies het verzoek dat u wilt invoeren:



In dit voorbeeld wordt querySubscriberApplication gebruikt.

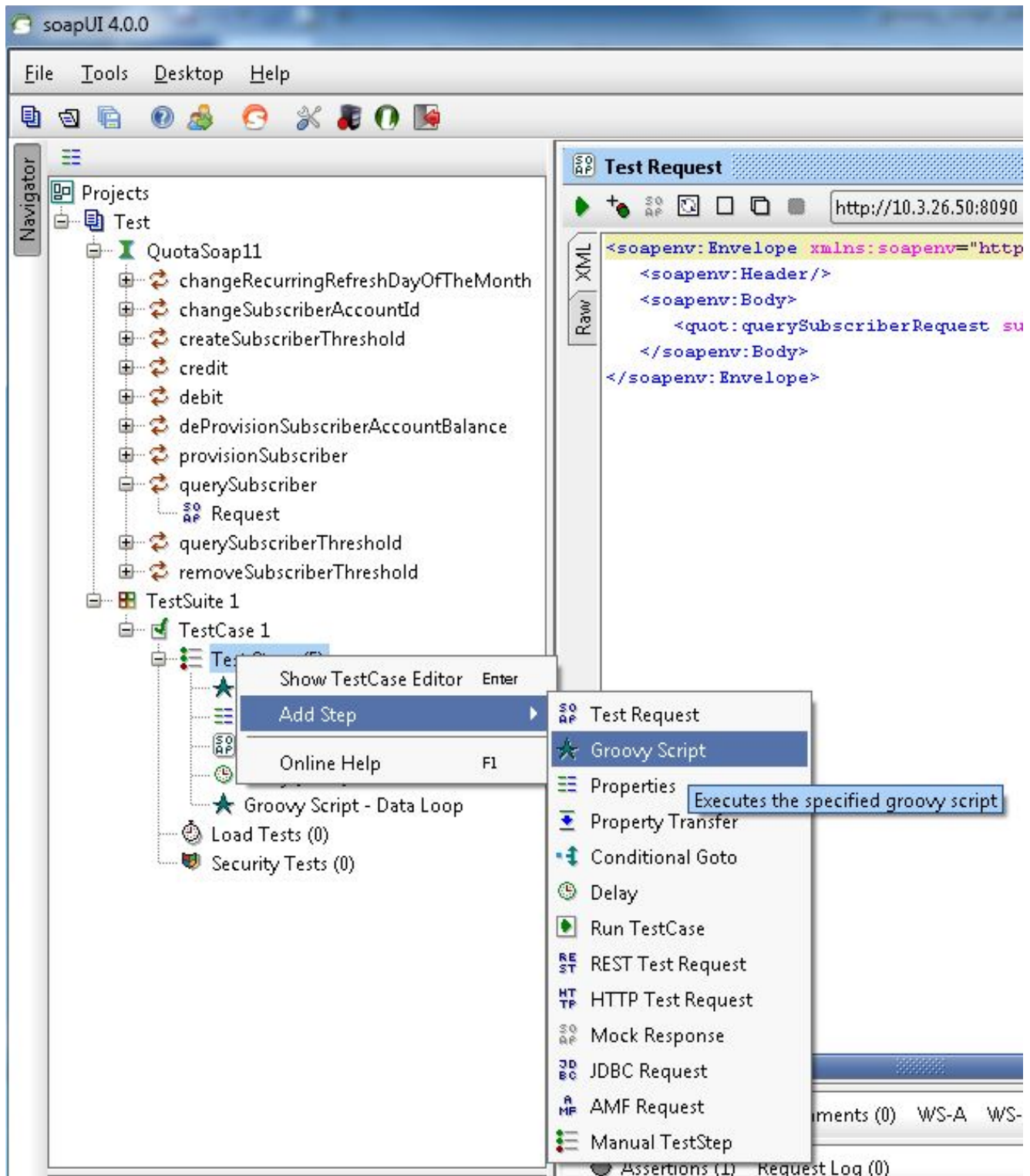


8. In het verzoek vervangt de uitbreidingscode de veldwaarden van wat u vraagt. In dit voorbeeld de ? van de **SubscriberID=?** in querySubscriberApplication wordt vervangen met uitbreidingscode **`{Eigenschap - Looper#Value}`** (soap_test_req_expansie_code):



Eigenschap - Looper is de naam van de Eigenschap TestStep die eerder is gemaakt en **Value** houdt de huidige abonnee-ID bij die wordt gelezen uit het bestand van abonnee-ID's.

9. Klik met de rechtermuisknop op de Test Stap en kies **Stap > Groovy Script** en kies de **datalink**:



10. Plakt deze code in de 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. In deze voorbeeldprocedure wordt een vertraging van 1000 ms tussen elke lus toegevoegd. Deze stap is optioneel. Met de vertraging zijn er nu vijf teststappen:

The screenshot shows a testing tool interface. On the left, a 'Navigator' pane displays a tree view of projects. Under 'Test', there is a 'QuotaSoap11' folder containing various test steps like 'changeRecurringRefreshDayOfTheMonth', 'changeSubscriberAccountId', etc. Below it is 'TestSuite 1', which contains 'TestCase 1'. 'TestCase 1' has five test steps: 'Groovy Script - DataSource', 'Property - Looper (5)', 'Test Request', 'Delay [1000]', and 'Groovy Script - Data Loop'. The 'Delay [1000]' step is highlighted in blue.

On the right, the 'TestCase 1' window is open. It shows a toolbar with icons for running, stopping, and other actions. Below the toolbar, the 'TestSteps' pane lists the five steps: 'Groovy Script - DataSource', 'Property - Looper (5)', 'Test Request', 'Delay [1000]', and 'Groovy Script - Data Loop'. The 'Delay [1000]' step is highlighted in blue.

At the bottom, there are two tabs: 'TestCase Properties' and 'Test Properties'. The 'Test Properties' tab is active, showing a table with the following data:

Property	Value
Name	TestCase 1

At the bottom right, there is a console window showing the following output:

```
at java.net.PlainSocketImpl.connectToAddress(Unknown S
at java.net.PlainSocketImpl.connect(Unknown Source)
at java.net.SocksSocketImpl.connect(Unknown Source)
```

12. Klik op de knop groen **uitvoeren** om de vijf Test-stappen in het venster TestCase te uitvoeren.