

Automatice las API con Groovy Script

Contenido

[Introducción](#)

[Crear un proyecto soapUI](#)

[Crear una solicitud API soapUI](#)

[Crear un caso de prueba de soapUI](#)

Introducción

Este documento describe cómo crear una solicitud de la interfaz de programadores de aplicaciones (API) de soapUI y cómo crear un caso de prueba de soapUI que se ejecuta en bucle sobre los pasos de prueba que automatizan las solicitudes de API a Quantum Policy Suite (QPS).

El ejemplo de caso de prueba soapUI en este artículo implementa Pasos de prueba que leen un archivo de ID de suscriptores y luego crean y envían una querySubscriberRequest a QPS.

Crear un proyecto soapUI

Antes de iniciar este procedimiento, instale la aplicación soapUI en el escritorio. Puede descargar el ejecutable de instalación de soapUI desde www.soapui.org.

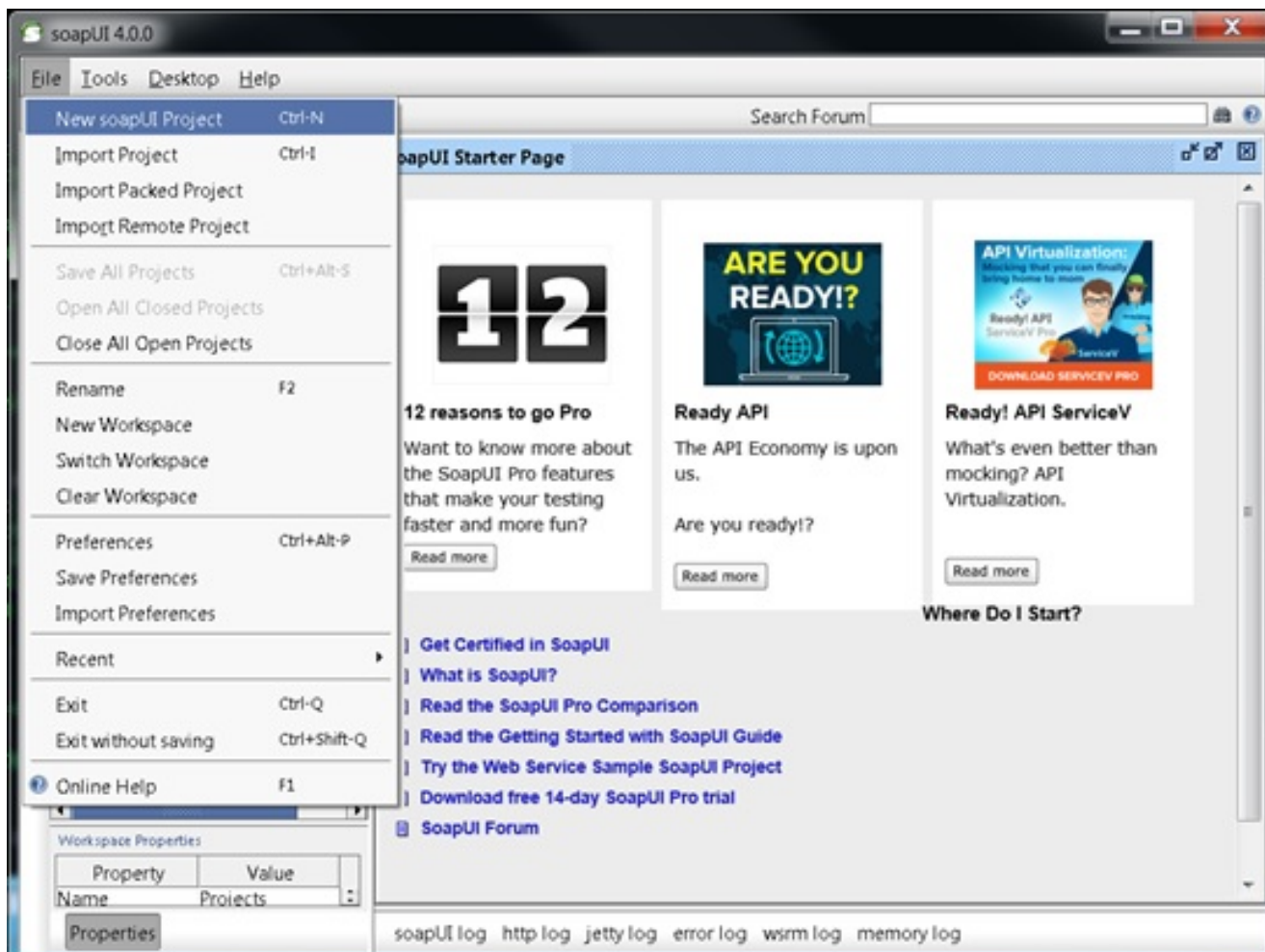
Antes de crear una solicitud de API o un caso de prueba, debe crear un proyecto soapUI. Necesita el archivo WSDL (Lenguaje de descripción de servicios Web) y el archivo XML Schema Description (XSD) para crear el proyecto. WSDL especifica las API admitidas. Normalmente, puede obtener WSDL y XSD de QPS cuando ejecuta estos comandos desde el equilibrador de carga (LB):

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

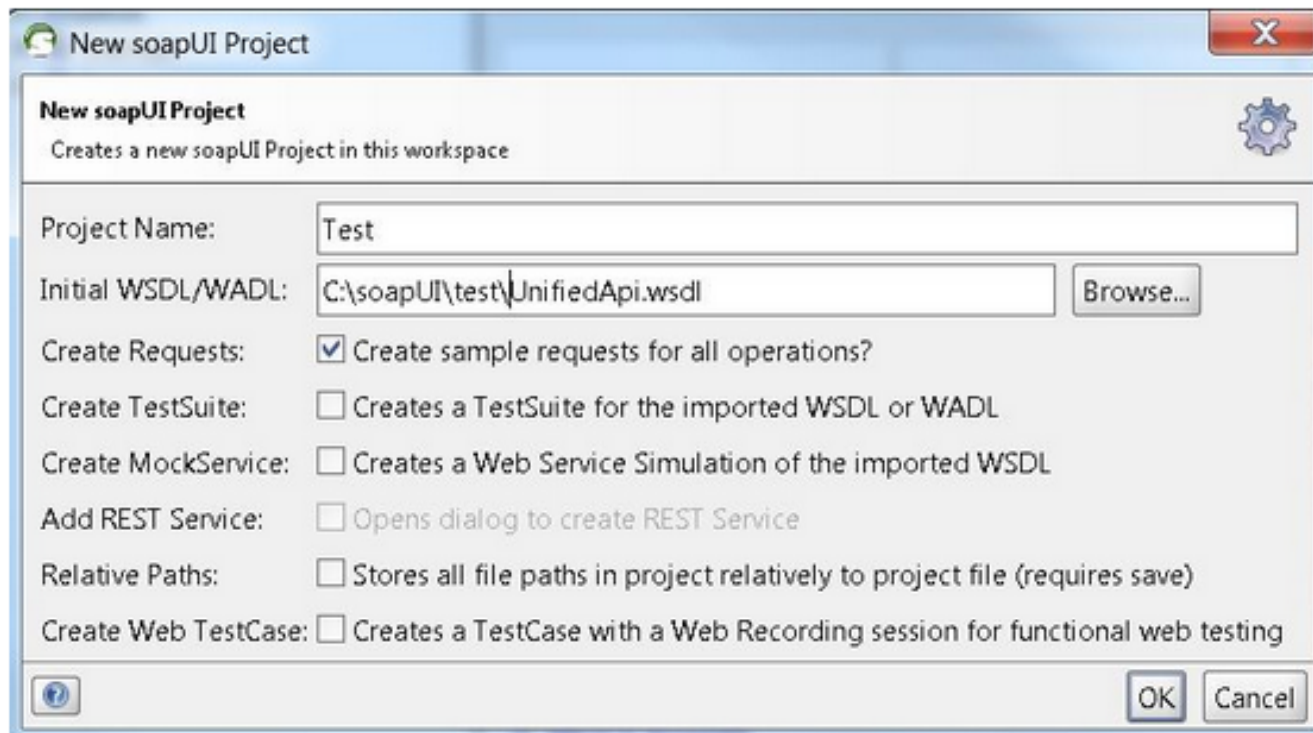
Almacene WSDL y XSD en el mismo directorio en el escritorio donde tiene previsto ejecutar la aplicación soapUI.

Complete estos pasos para crear el proyecto soapUI:

1. Elija **File > New soapUI Project** en la ventana soapUI:



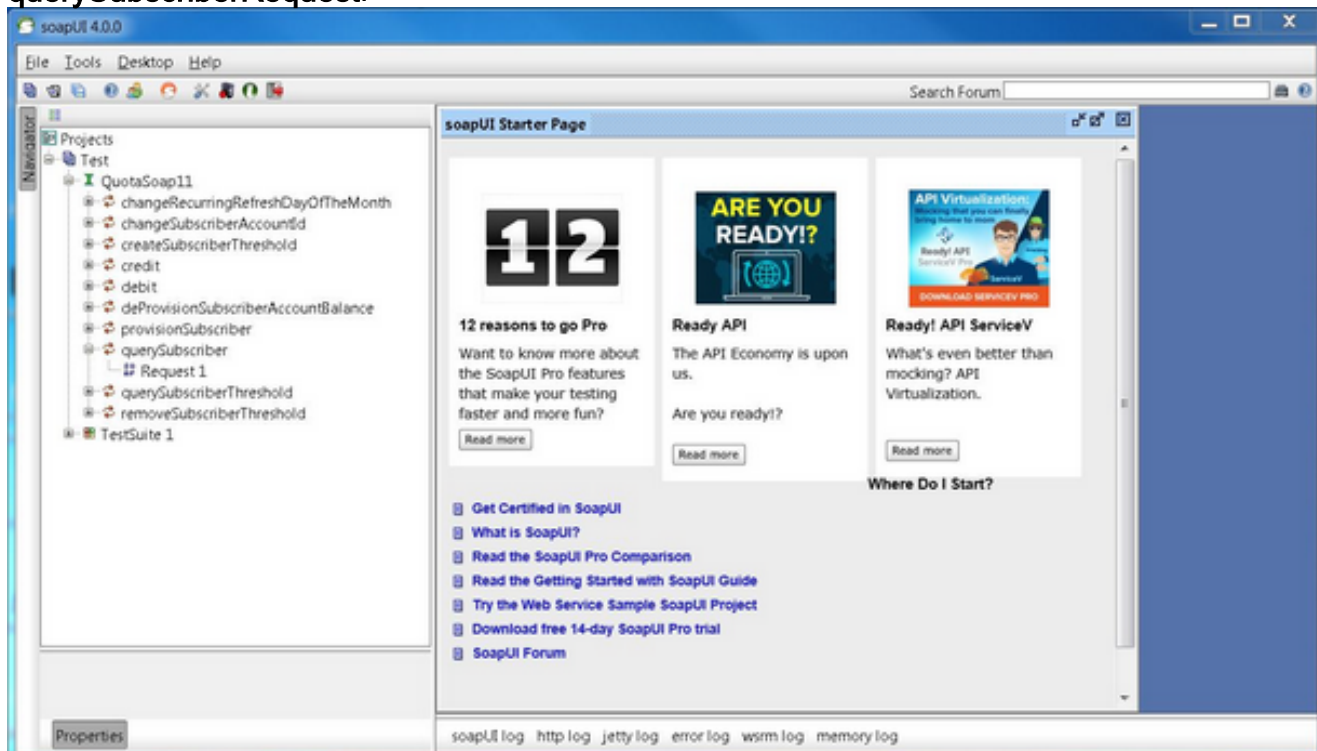
2. En la ventana Nuevo proyecto soapUI, introduzca un nombre para el proyecto en el campo Nombre del proyecto e introduzca la ubicación en la que se almacena el archivo WSDL en el campo WSDL/WADL inicial. Haga clic en **Aceptar** cuando haya terminado.



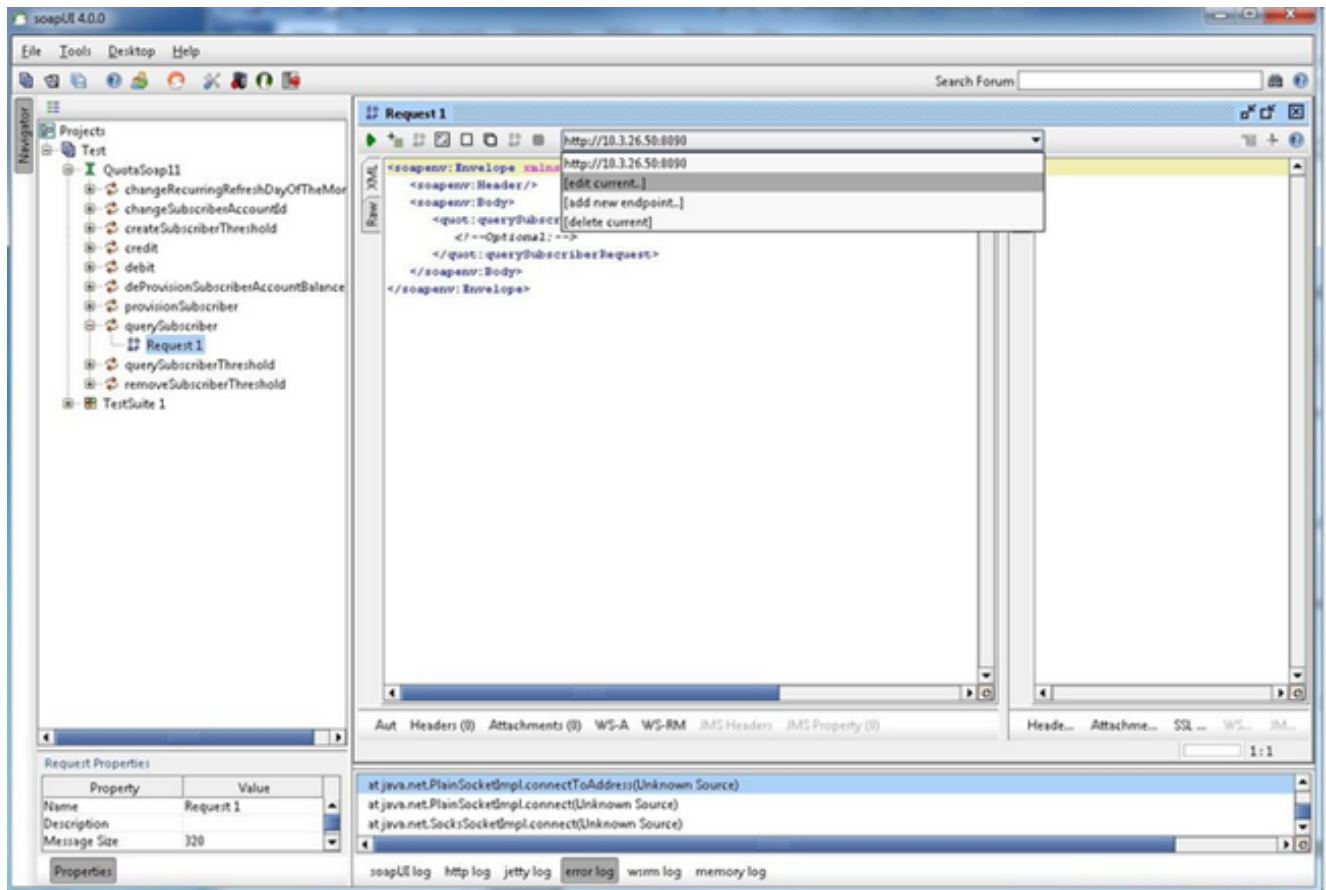
Crear una solicitud API soapUI

Complete estos pasos para crear una solicitud de API soapUI:

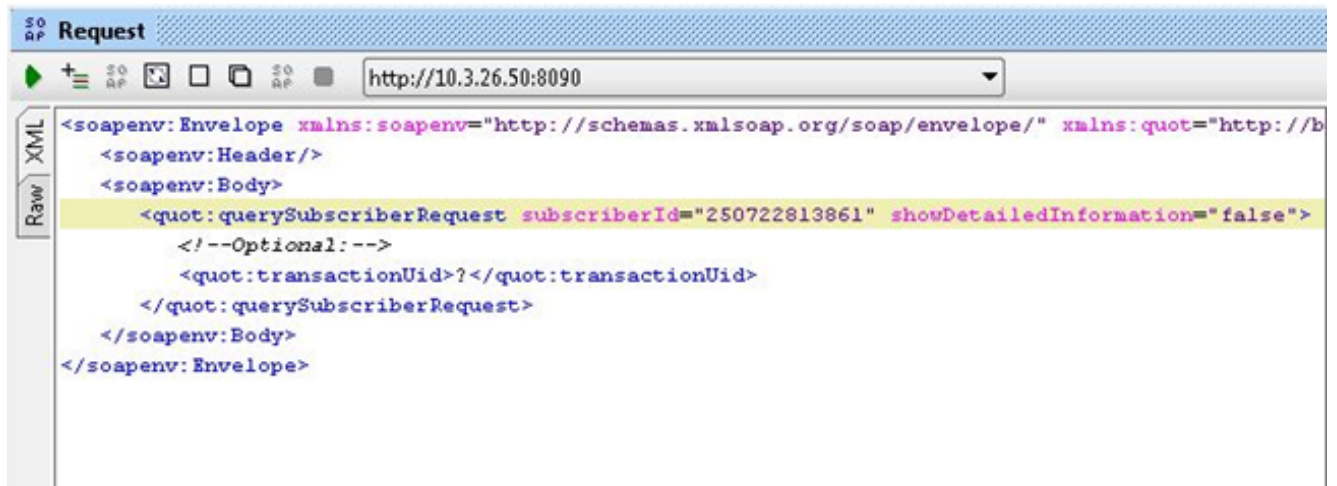
1. Expanda el proyecto soapUI que creó para ver las API. También puede expandir una de las API para ver la solicitud. En este ejemplo, se expande **querySubscriberRequest**:



2. Abra la solicitud para ver la ventana de solicitud con el XML que forma la consulta. En la ventana Request (Solicitud), edite la dirección IP `http://` a la dirección IP y al puerto. Normalmente, ésta es la dirección IP `lbvip01` y el puerto donde desea enviar la solicitud, como muestra este ejemplo:



3. Modifique los campos del XML con los datos que desea enviar en la solicitud. En este ejemplo, la solicitud es querySubscriberRequest. Modifique el ID de suscriptor para el suscriptor al que desea consultar y establezca showDetailInformation en false:



4. Haga clic en el botón verde **Ejecutar** en la parte superior de la ventana Solicitud para ejecutar la consulta.

Crear un caso de prueba de soapUI

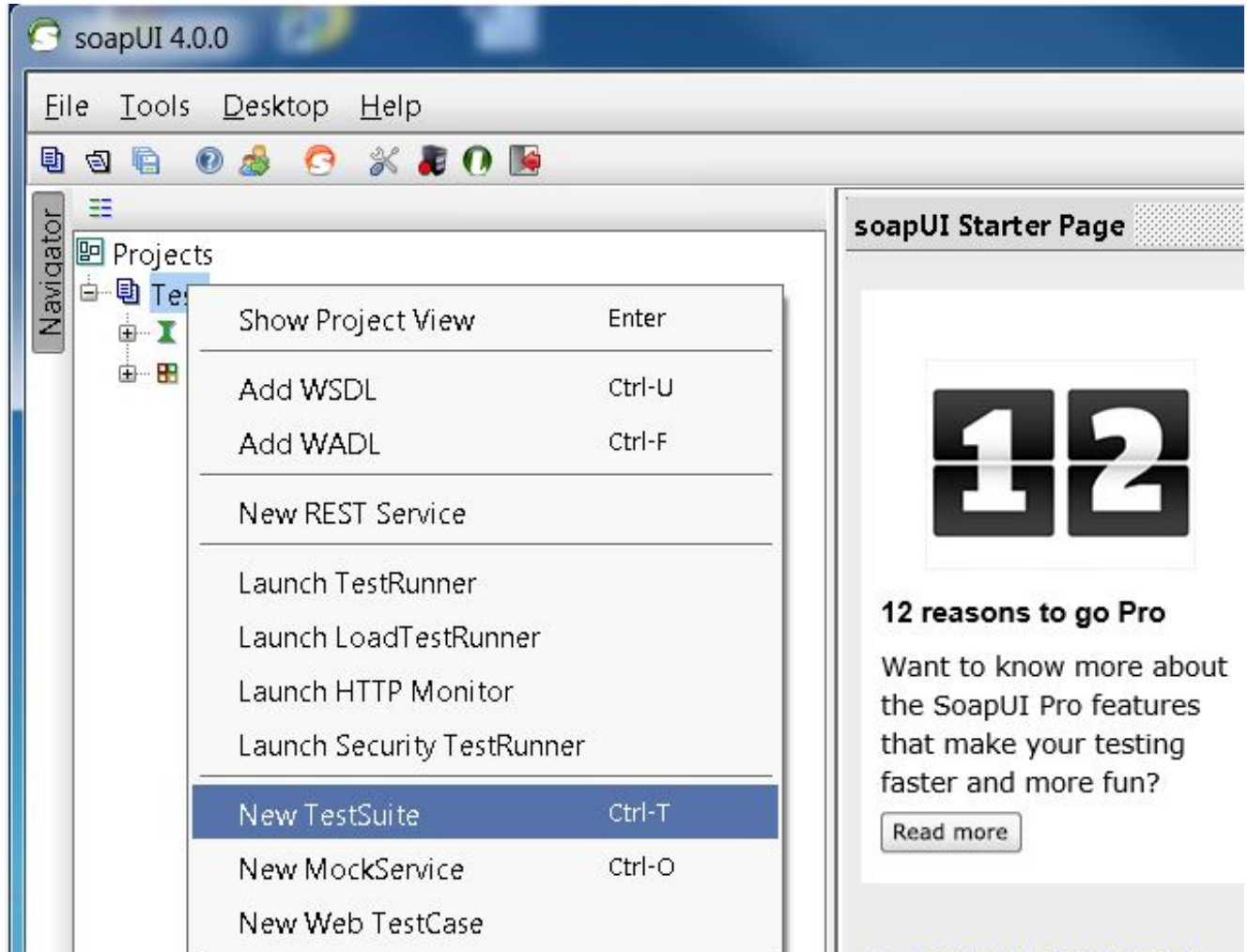
Este procedimiento explica cómo crear un conjunto de pruebas que pueda automatizar cuando se envíen API a QPS.

En este procedimiento de ejemplo, el conjunto de pruebas realiza un bucle sobre una lista de ID de suscriptores y luego utiliza esos ID de suscriptores en querySubscriberRequest que envía a QPS. La lista de ID de suscriptores se encuentra en una sola línea en un archivo de texto llamado

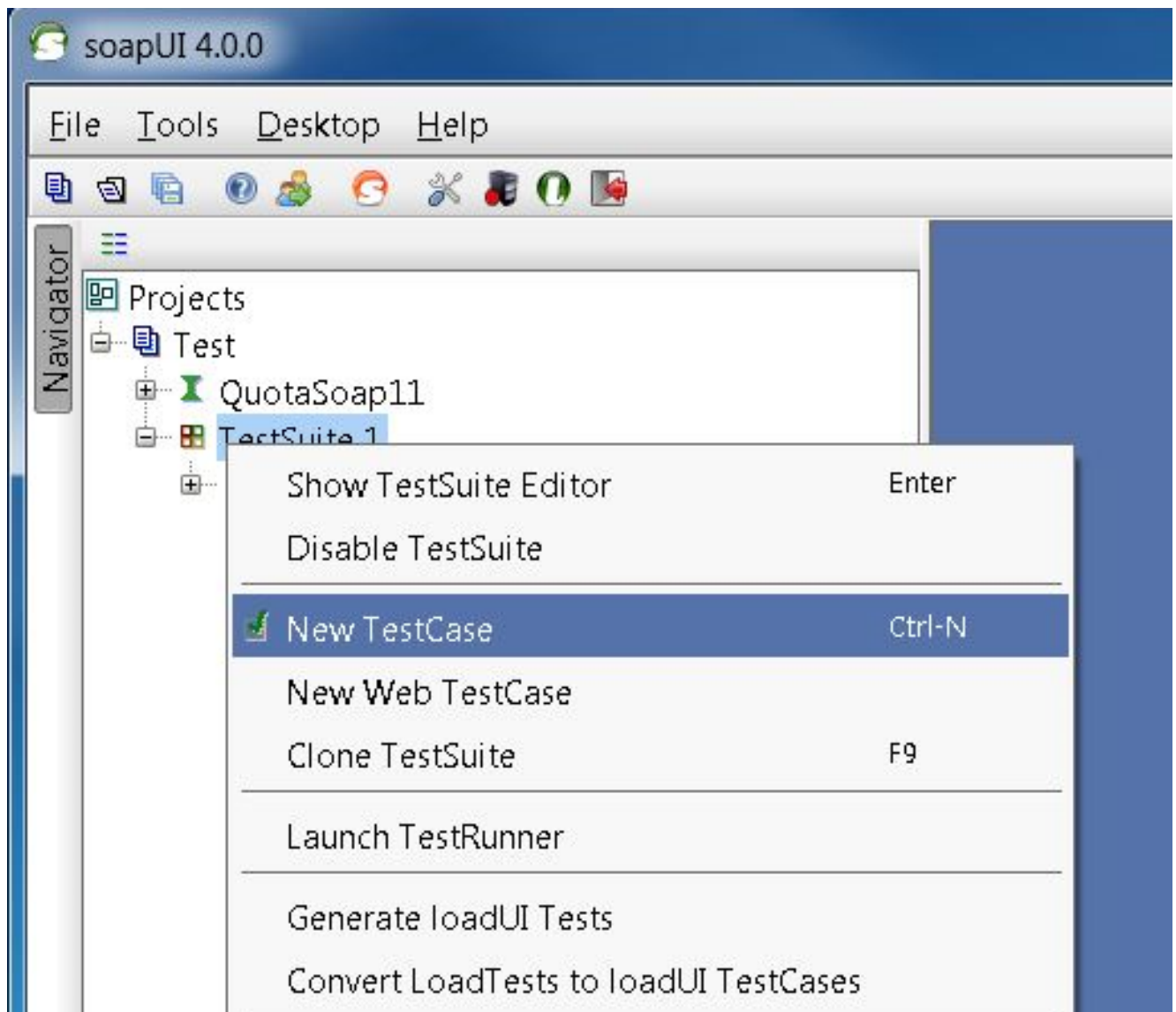
subid.txt.

Complete estos pasos para crear el conjunto de pruebas:

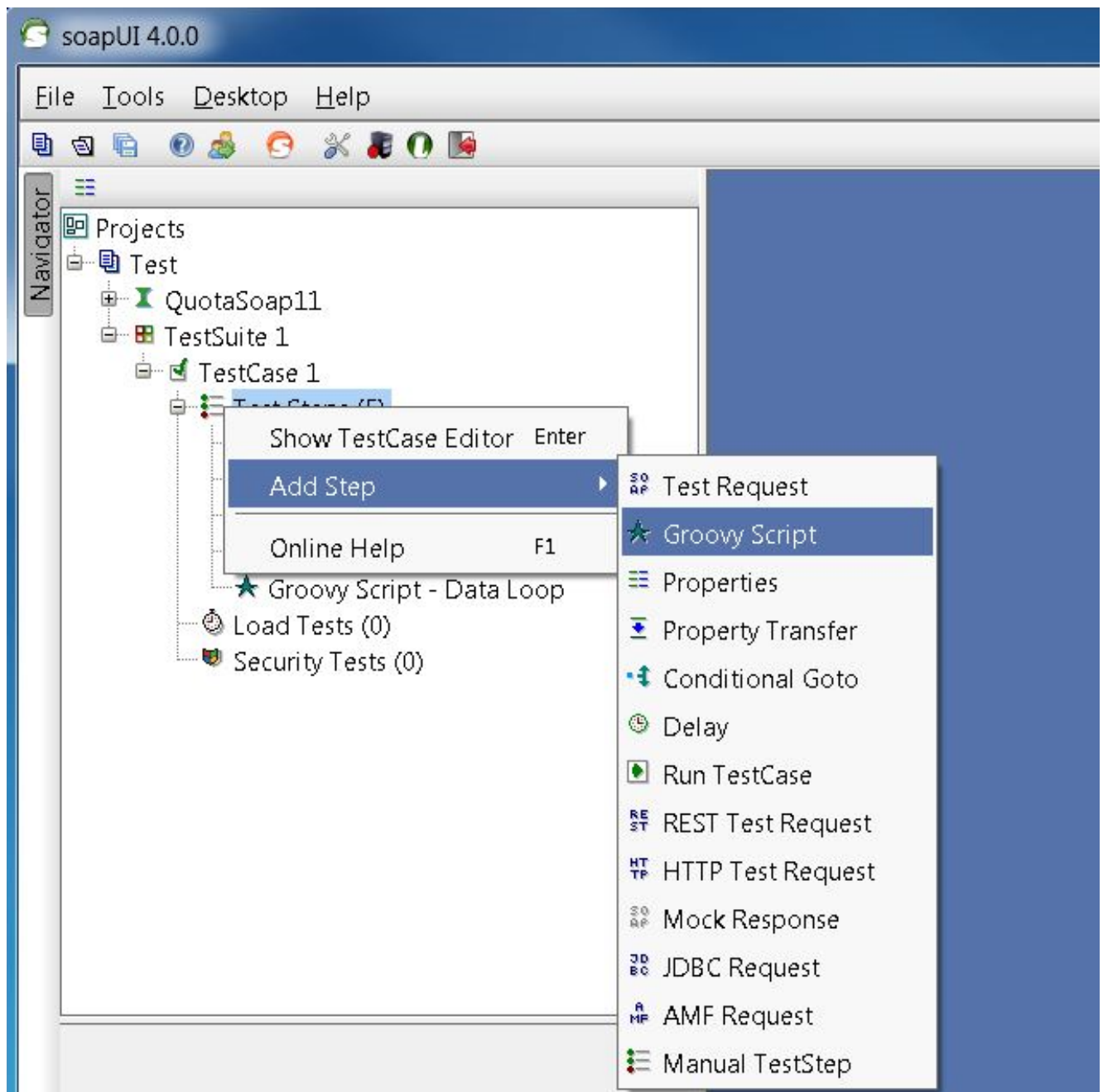
1. En el proyecto soapUI que creó, cree un nuevo conjunto de pruebas. Haga clic con el botón derecho del ratón en la interfaz de usuario del jabón y elija **New TestSuite**.



2. Haga clic con el botón derecho del ratón en Test Suite y elija **New TestCase**.



3. Haga clic con el botón derecho del mouse en el caso de prueba y elija **Add Step > Groovy Script** para agregar un paso de prueba Groovy Script. Denomínelo **Origen de Datos**:

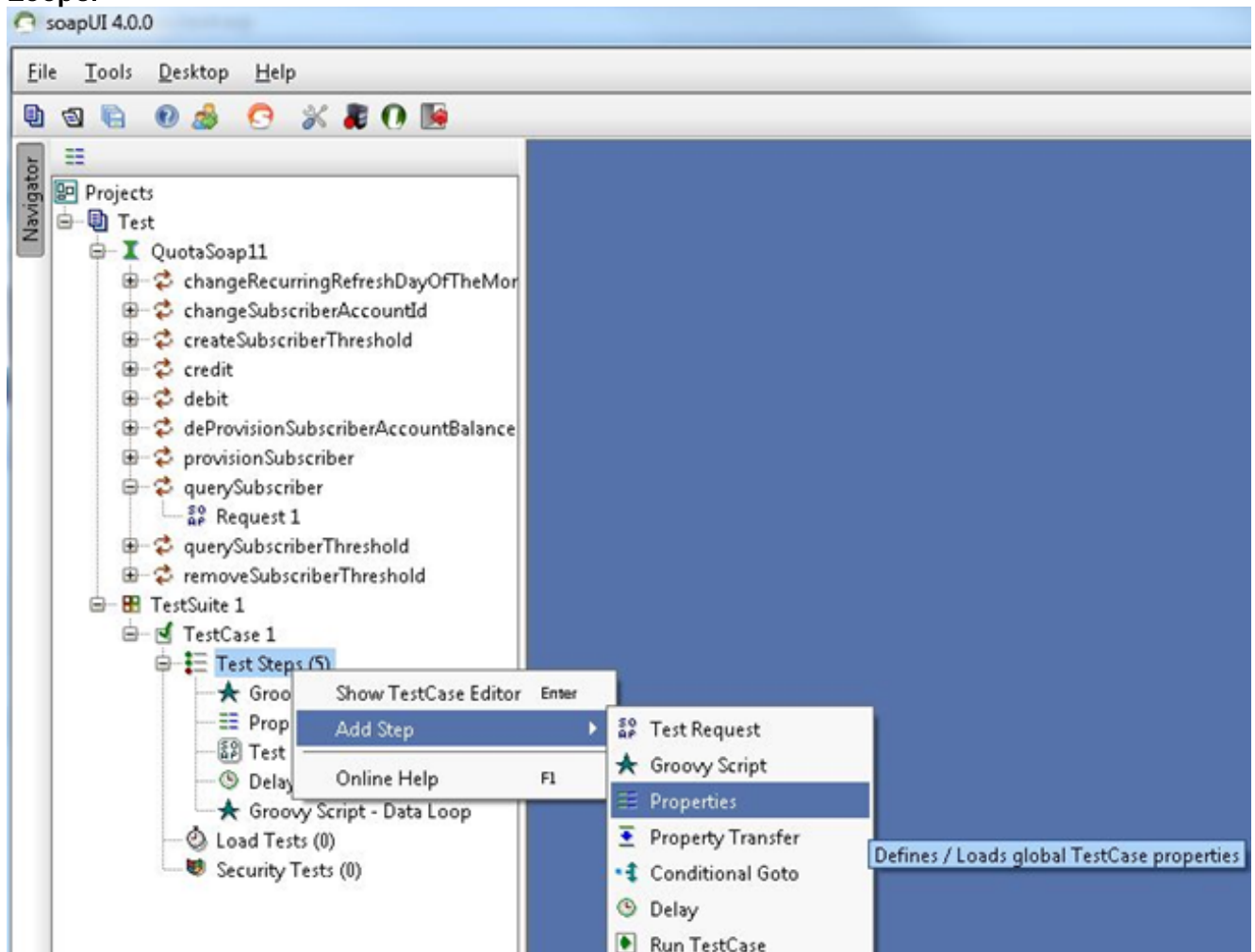


4. En el archivo Origen de datos, pegue este código. Este código lee el archivo **C:/subid.txt** que contiene un ID de suscriptor en cada línea:

```
import com.eviware.soapui.support.XmlHolder def myTestCase = context.testCase
def counter,next,previous,sizeFile tickerEnumFile = new File("C:/subid.txt") //subscriber
IDs separed 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. Haga clic con el botón derecho del mouse en el paso de prueba y elija **Agregar paso > Propiedades** para agregar un paso de prueba de Propiedad y asignarle el nombre **Propiedad**

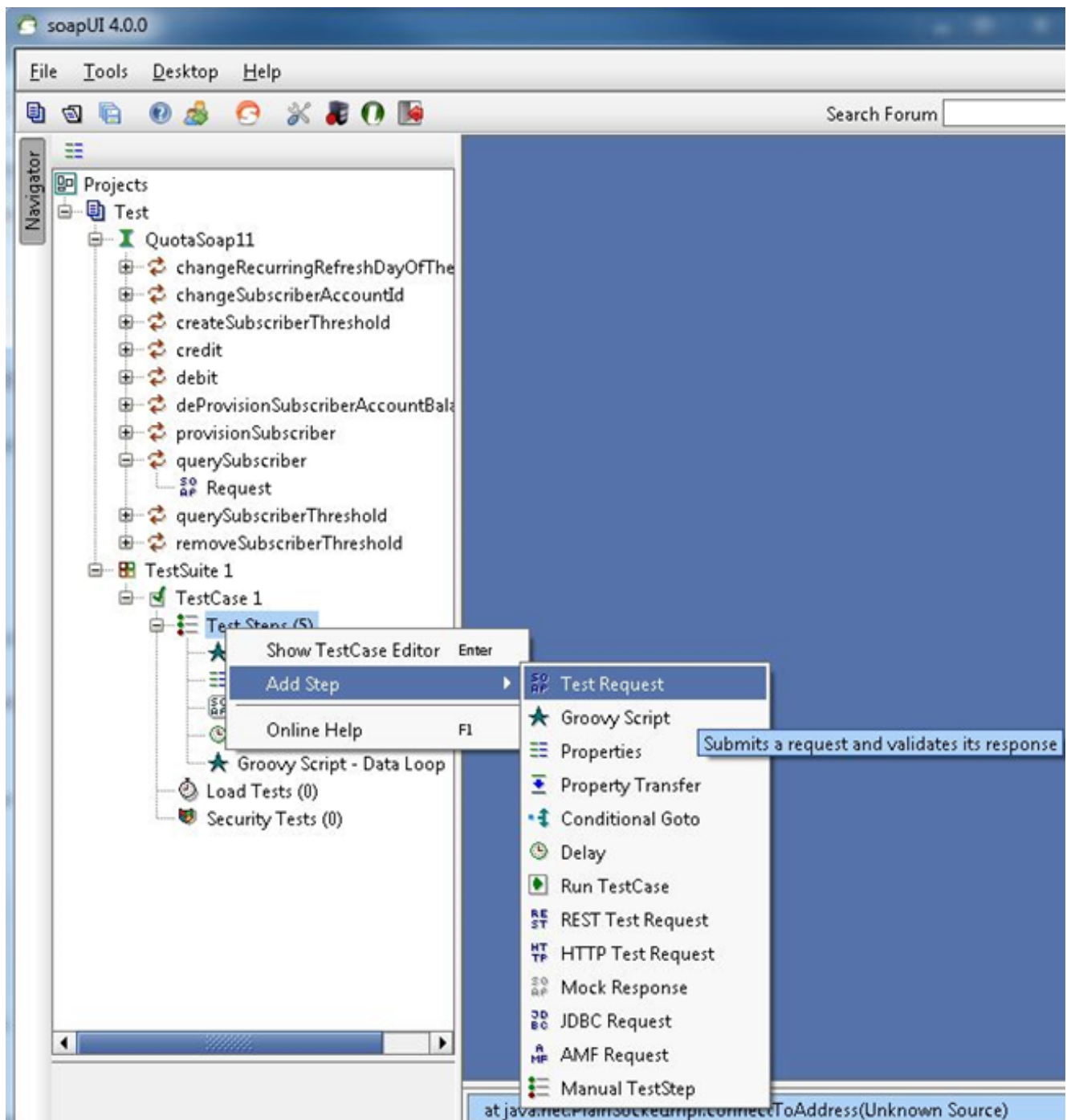
Looper.



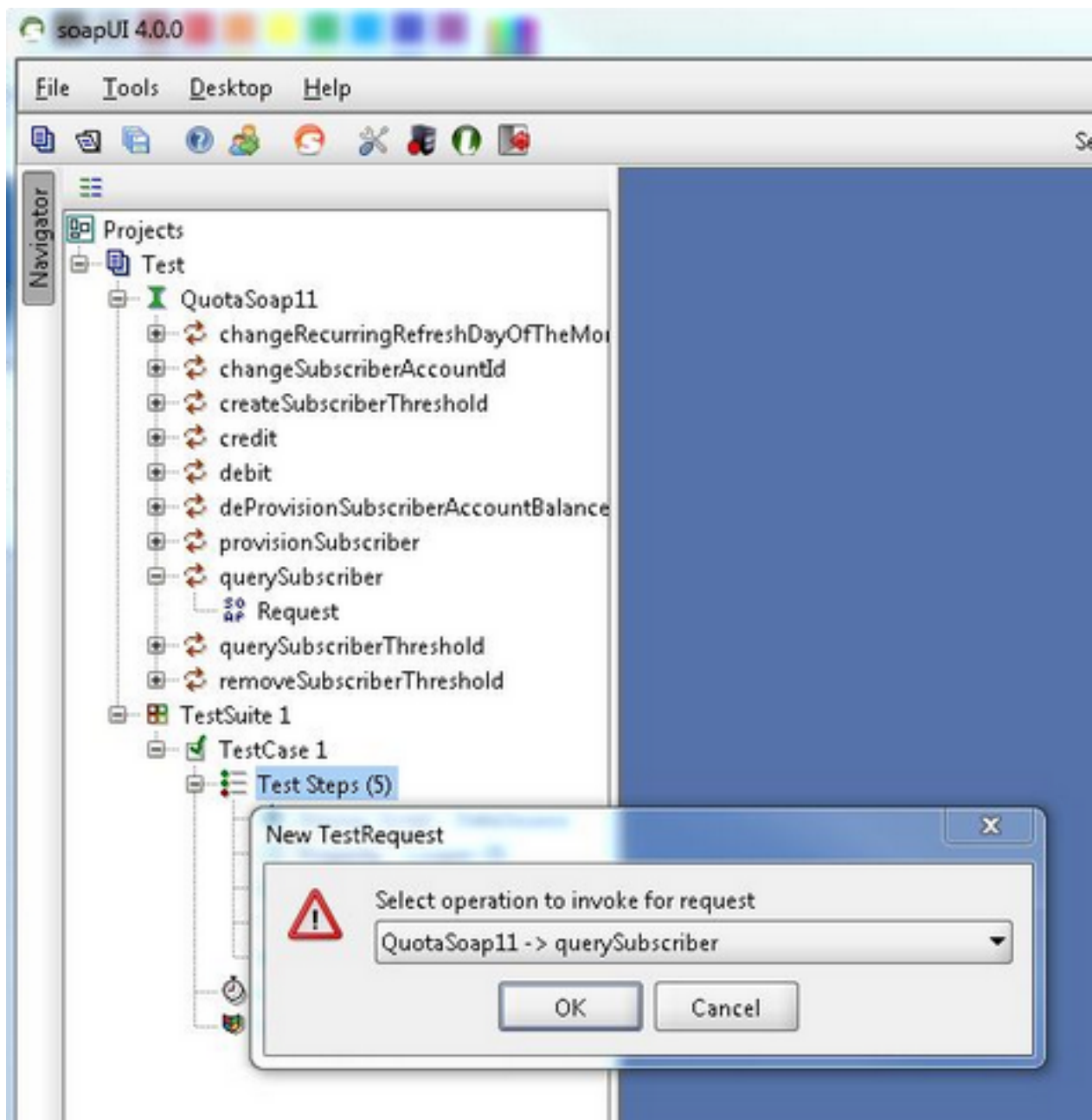
6. Agregue estas propiedades definidas por el usuario del paso de prueba Looper: TotalValor: en nuestro ejemplo, se encuentra en espera la ID del suscriptor leída del archivo IDs del suscriptorCuentaSiguienteStopLoop

Name	Value
Total	
Value	
Count	
Next	
StopLoop	

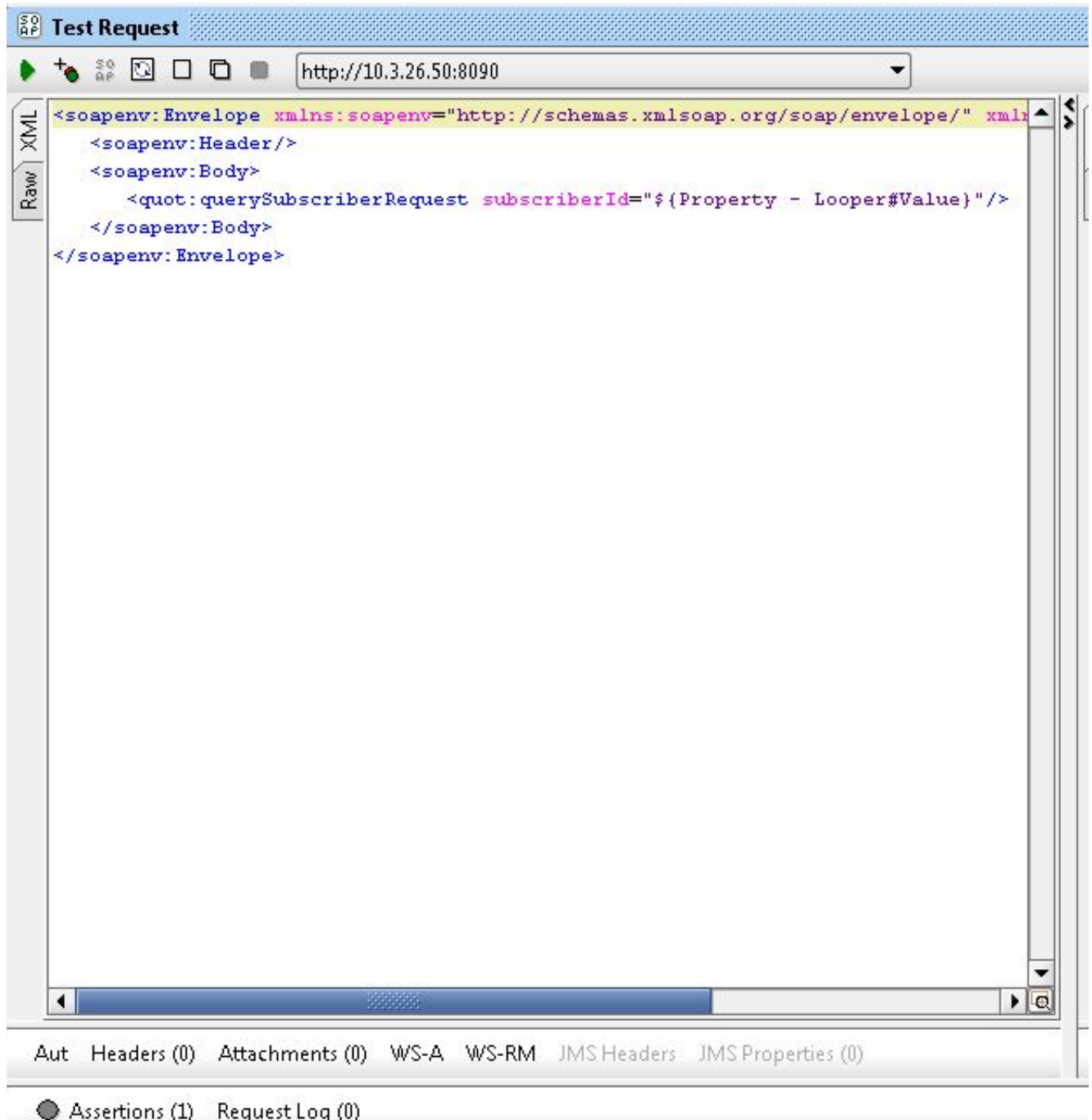
7. Haga clic con el botón derecho en el paso de prueba y elija **Agregar paso > Solicitud de prueba** para agregar un paso de prueba de solicitud de prueba y elija la solicitud que desea invocar:



En este ejemplo, se utiliza querySubscriberRequest.

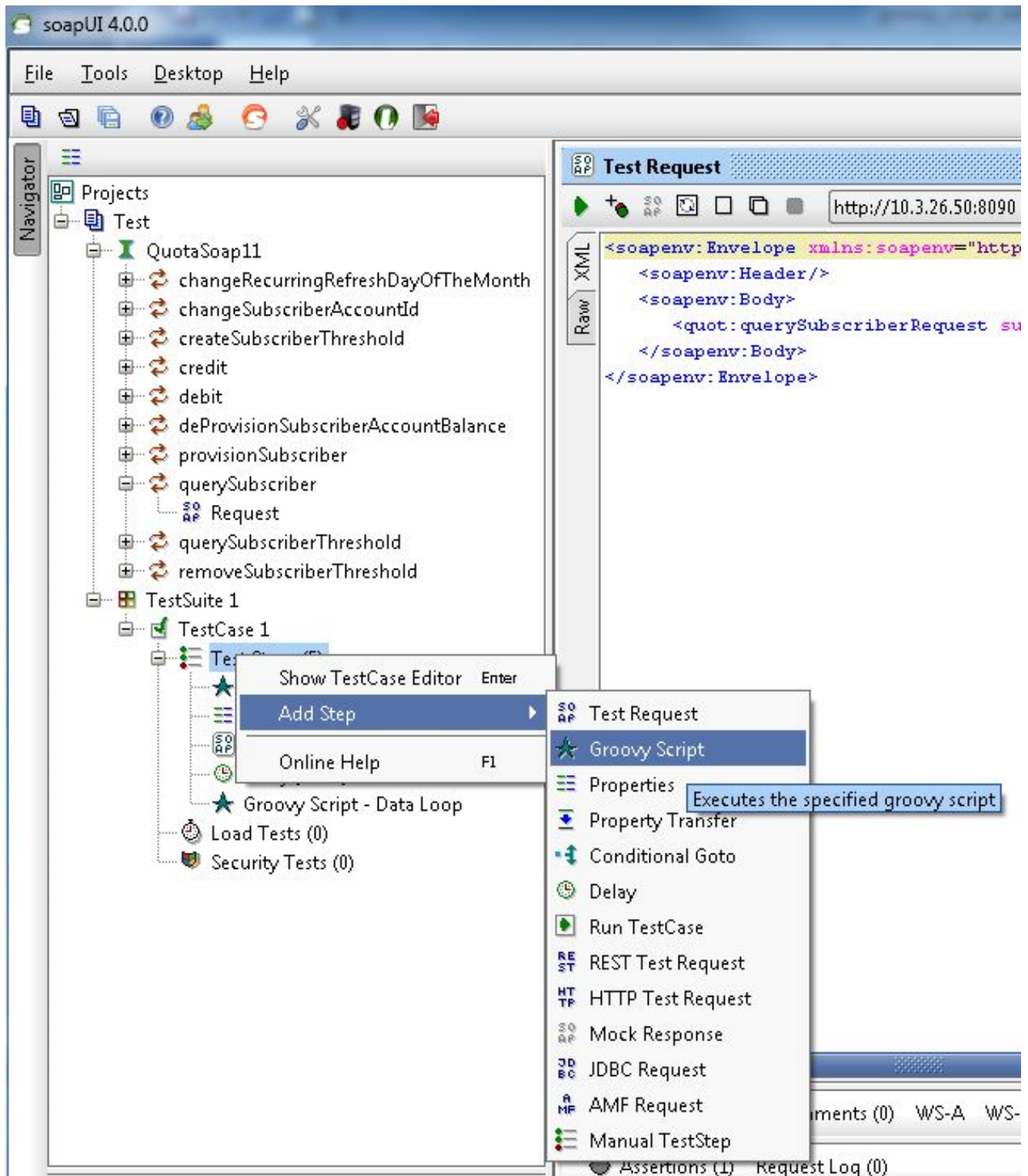


8. En la solicitud, el código de expansión reemplaza los valores de campo de lo que consulta. ¿En este ejemplo el ? del **SubscriberId=?** en querySubscriberRequest se reemplaza por el código de expansión **\${Property - Looper#Value}** (soap_test_req_expansion_code):



Propiedad: Looper es el nombre de Property TestStep creado anteriormente y Value contiene la ID de suscriptor actual leída del archivo de ID de suscriptor.

9. Haga clic con el botón derecho del ratón en el paso de prueba y elija **Agregar paso > Groovy Script** y denomínelo **Bucle de datos**:



10. Pegue este código en el bucle de datos de secuencias de comandos Groovy:

```

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. En este procedimiento de ejemplo, se agrega una demora de 1000 ms entre cada loop.
This step is optional. Con el retraso, ahora hay cinco pasos de prueba:

The screenshot displays the JMeter interface. On the left, the Navigator pane shows a tree structure under 'Test' with a 'TestSuite 1' containing 'TestCase 1'. 'TestCase 1' has five test steps: 'Groovy Script - DataSource', 'Property - Looper (5)', 'Test Request', 'Delay [1000]', and 'Groovy Script - Data Loop'. The right pane shows 'TestSteps' for 'TestCase 1' with the same five steps. Below the TestSteps pane is a table for 'TestCase Properties' and 'Test Properties'.

Property	Value
Name	TestCase 1

At the bottom, the execution log shows the following stack trace:

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

12. Haga clic en el botón verde **Run** para ejecutar los cinco pasos de prueba en la ventana TestCase.