

Groovy عم تاقى بطلت لة جمر رب تاهجاو ةتمت أ Script

المحتويات

[المقدمة](#)

[إنشاء مشروع SoapUI](#)

[إنشاء طلب واجهة برمجة تطبيقات SoapUI](#)

[إنشاء حالة اختبار SoapUI](#)

المقدمة

يوضح هذا المستند كيفية إنشاء طلب واجهة برمجة تطبيقات (API) SoapUI (وكيفية إنشاء حالة اختبار SOAPui التي تقوم بتكرار خطوات الاختبار التي تقوم بآتمتة طلبات API إلى مجموعة نهج الكم (QPS).

تقوم حالة اختبار soapUI في هذه المقالة بتنفيذ خطوات اختبار تقرأ ملف معرفات المشترك ثم تقوم بإنشاء وإرسال استعلام SubscriberRequest إلى QPS.

إنشاء مشروع SoapUI

قبل بدء هذا الإجراء، قم بتثبيت تطبيق soapUI على سطح المكتب. يمكنك تنزيل ملف تثبيت SOAPui القابل للتنفيذ من www.soapui.org.

قبل إنشاء طلب واجهة برمجة تطبيقات أو "حالة اختبار"، يجب أولاً إنشاء مشروع soapUI. تحتاج إلى ملف لغة وصف خدمات الويب (WSDL) وملف وصف مخطط XML (XSD) لإنشاء المشروع. يحدد WSDL واجهات برمجة التطبيقات (API) المدعومة. أنت يستطيع عادة حصلت ال WSDL و XSD من ال QPS عندما أنت تشغل هذا أمر من التحميل موازن (LB):

• على العنوان <http://lbvip01:8080/ua/wsd/UnifiedApi.wsd>

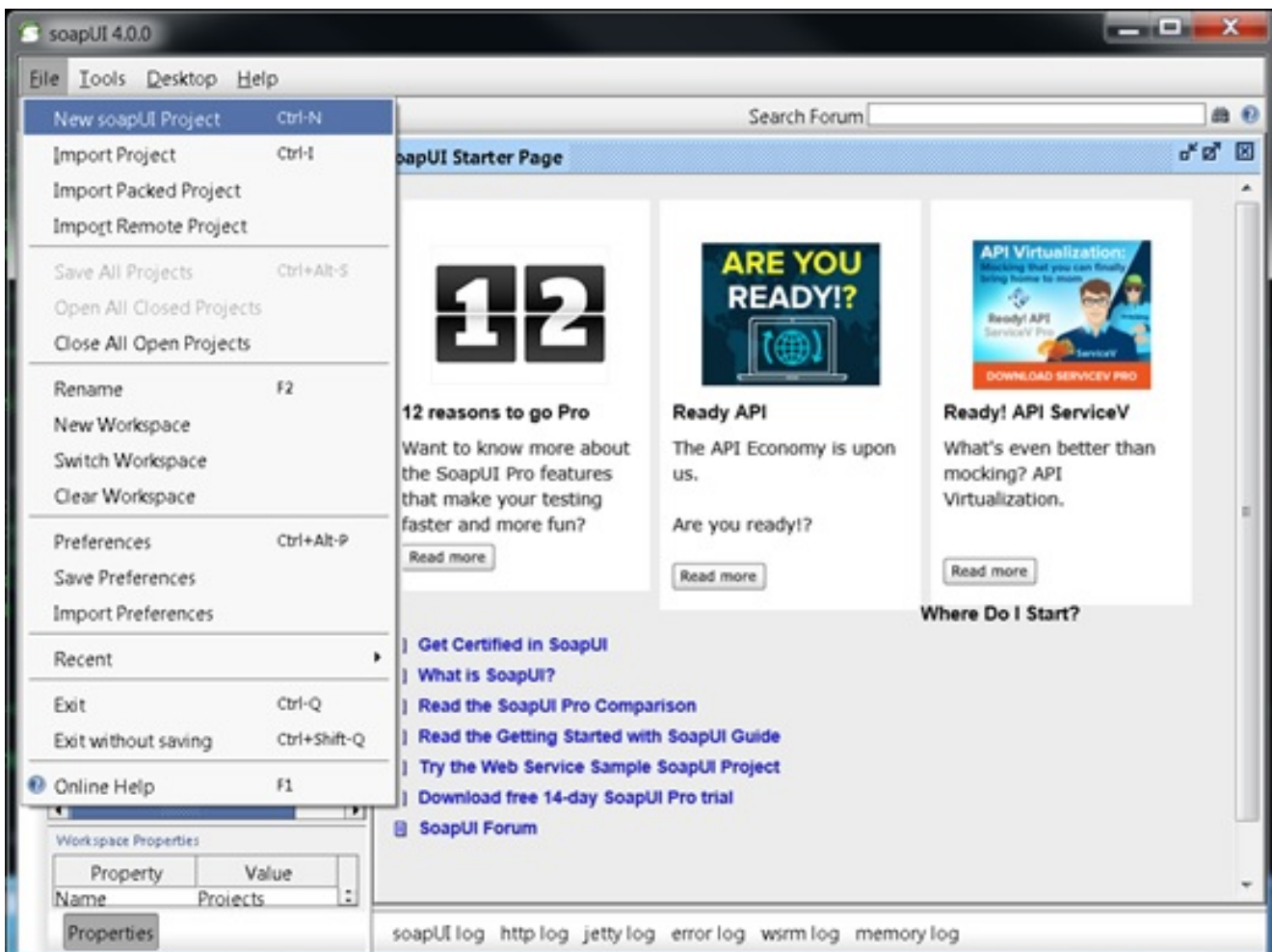
• على العنوان <http://lbvip01:8080/ua/wsd/UnifiedApi.xsd>

قم بتخزين WSDL و XSD في نفس الدليل على سطح المكتب حيث تخطط لتشغيل تطبيق soapUI.

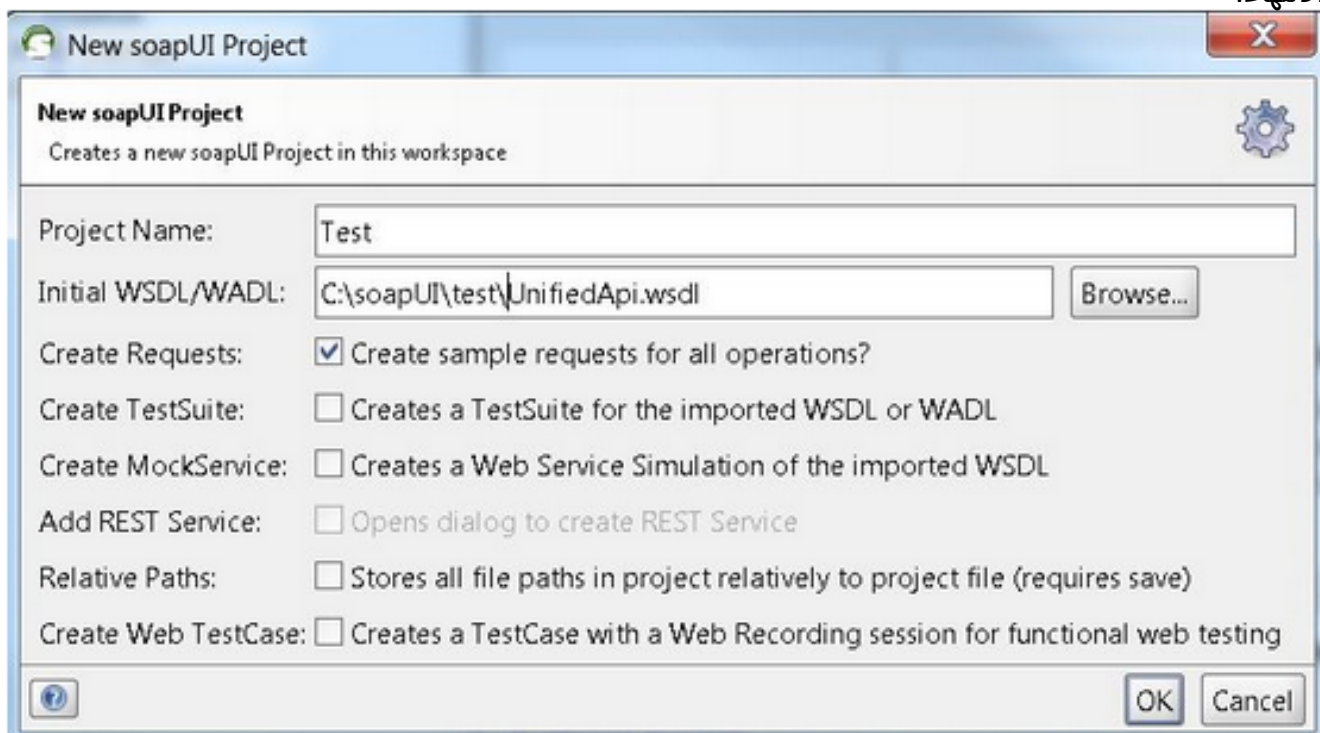
أكمل الخطوات التالية لإنشاء مشروع soapUI:

1. أختَر ملف < مشروع SoapUI جديد من نافذة

:soapUI



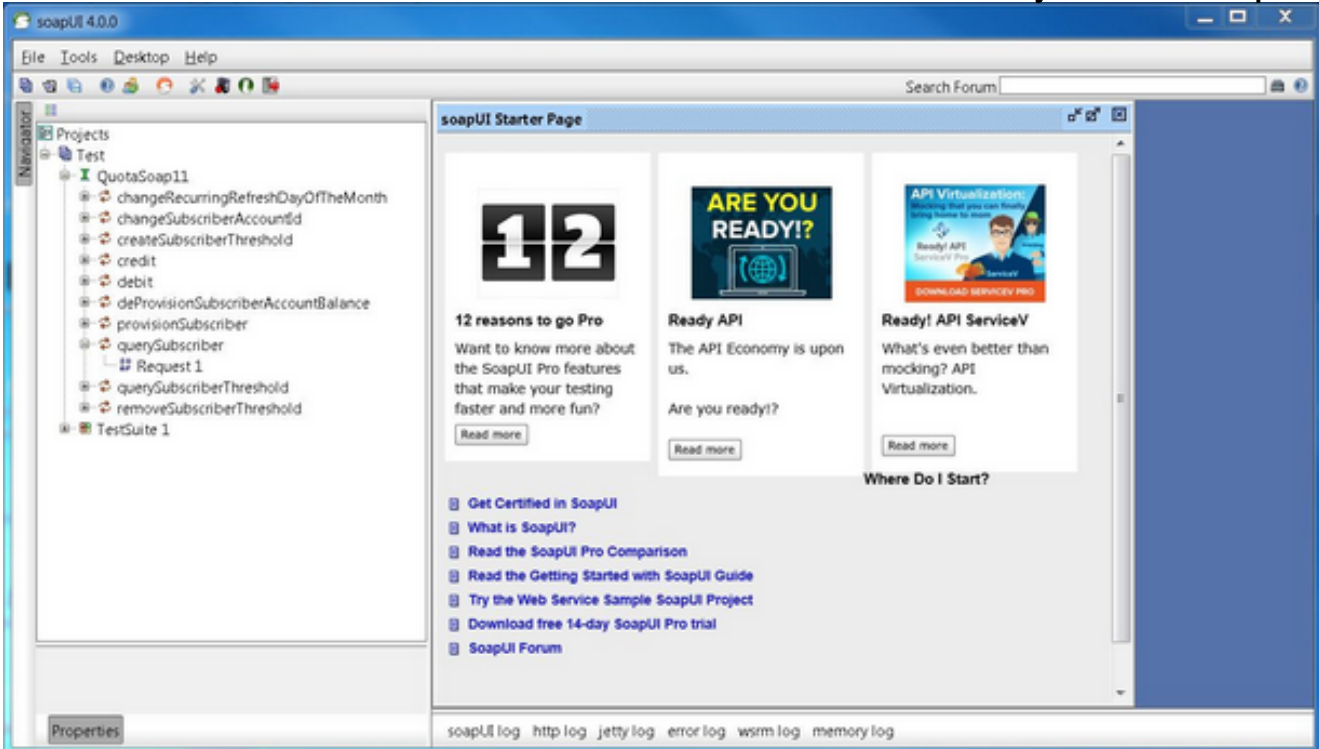
2. في نافذة مشروع SoapUI الجديدة أدخل اسم للمشروع في حقل اسم المشروع وأدخل الموقع الذي يتم فيه تخزين ملف WSDL في حقل WSDL/WADL الأولي. انقر فوق موافق عند الانتهاء.



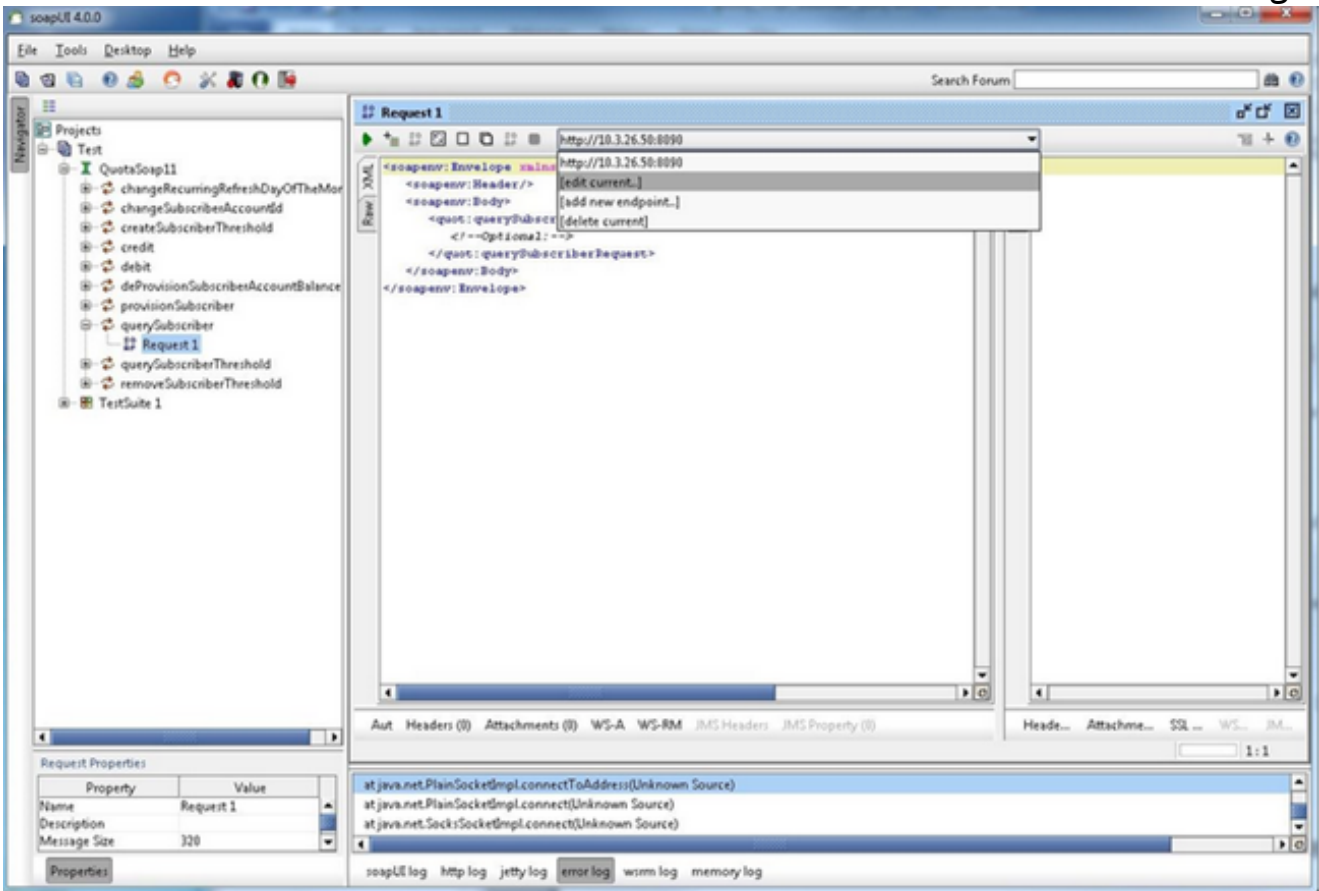
إنشاء طلب واجهة برمجة تطبيقات SoapUI

أكمل الخطوات التالية لإنشاء طلب واجهة برمجة تطبيقات SOAPui:

1. قم بتوسيع مشروع SoapUI الذي أنشأته لعرض واجهات برمجة التطبيقات. يمكنك أيضا توسيع أحد واجهات برمجة التطبيقات لعرض الطلب. في هذا المثال، يتم توسيع QuerySubscriberRequest:

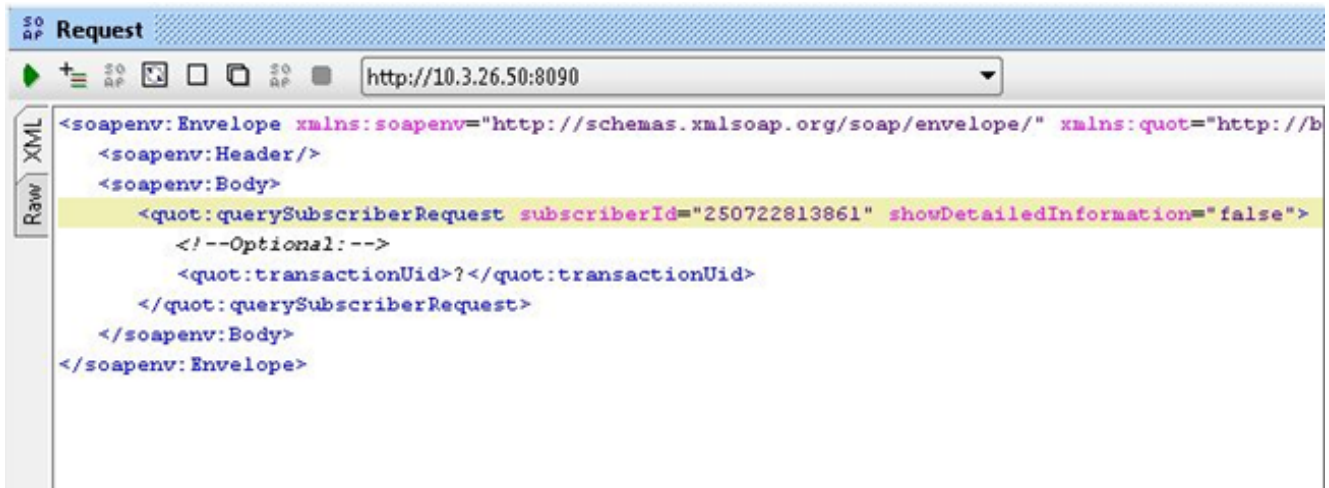


2. افتح الطلب لعرض نافذة الطلب مع XML الذي يشكل الاستعلام. في نافذة الطلب، قم بتحرير عنوان IP http:// إلى عنوان IP ومنفذ. هذا عادة ال lbvip01 عنوان ومنفذ حيث أنت تريد أن يرسل الطلب كما يوضح هذا مثال:



3. قم بتعديل الحقول الموجودة في XML باستخدام البيانات التي تريد إرسالها في طلبك. في هذا المثال، الطلب هو querySubscriberRequest. قم بتعديل معرف المشترك للمشارك الذي تريد الاستعلام عنه وقم بتعيين showDetailInformatin على

:false



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:quot="http://b
<soapenv:Header/>
<soapenv:Body>
  <quot:querySubscriberRequest subscriberId="250722813861" showDetailedInformation="false">
    <!--Optional:-->
    <quot:transactionUId?</quot:transactionUId>
  </quot:querySubscriberRequest>
</soapenv:Body>
</soapenv:Envelope>
```

4. انقر فوق الزر تشغيل أخضر في أعلى نافذة الطلب لتشغيل الاستعلام.

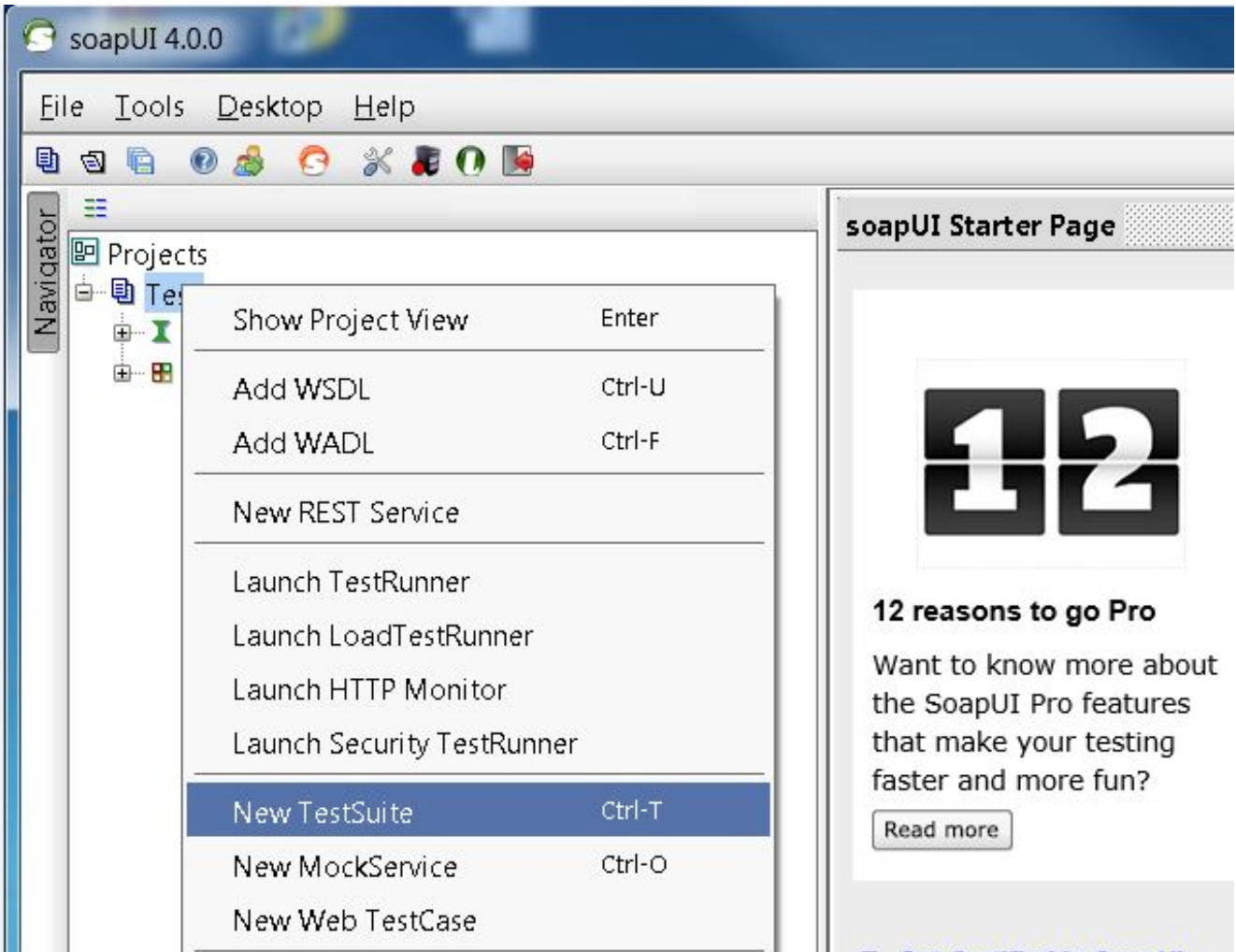
إنشاء حالة اختبار الSoapUI

يشرح هذا الإجراء كيفية إنشاء مجموعة اختبار يمكنها أتمتة عند إرسال وإجهات API إلى QPS.

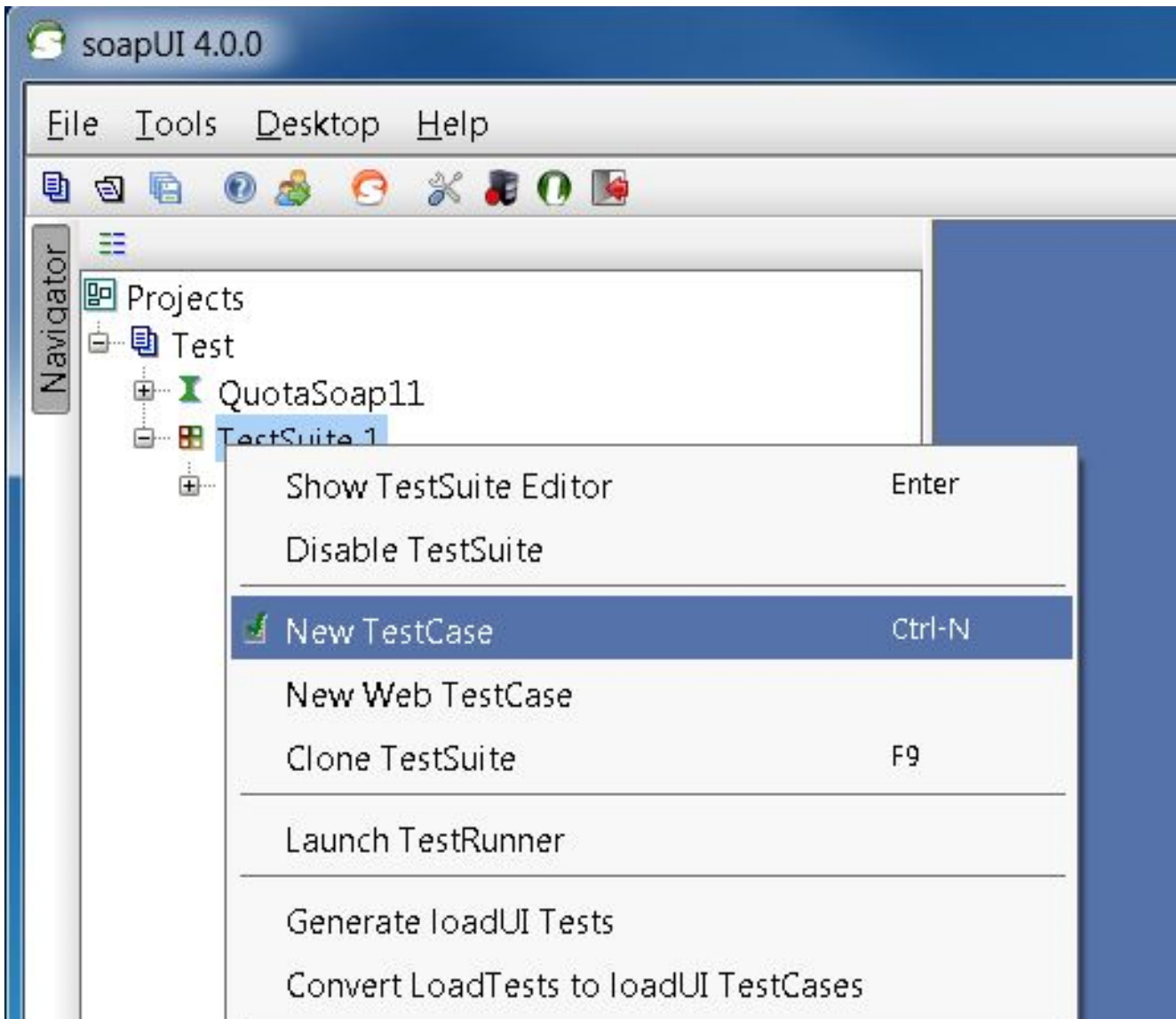
في إجراء المثال هذا، يتم إجراء مجموعة الاختبار عبر قائمة بمعرفات المشترك ثم يستخدم معرفات المشترك هذه في QuerySubscriberRequest التي يرسلها إلى QPS. قائمة معرفات المشترك موجودة في سطر واحد في ملف نصي يسمى `subid.txt`.

أتمت هذا steps in order to خلقت الاختبار مجموعة:

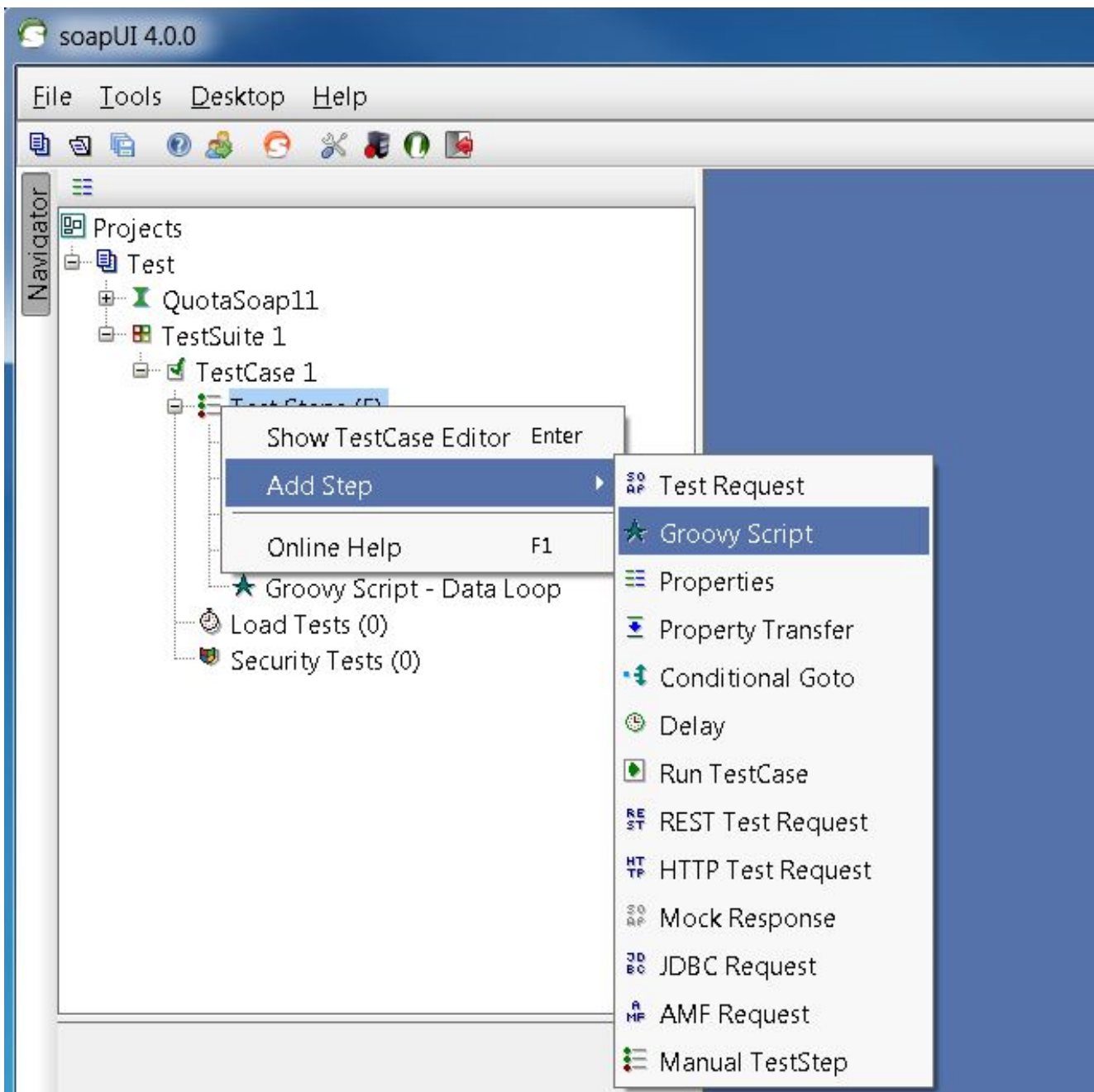
1. في مشروع SoapUI الذي أنشأته، قم بإنشاء مجموعة اختبار جديدة. انقر بزر الماوس الأيمن على SoapUI واختر `TestSuite` جديد.



2. انقر بزر الماوس الأيمن فوق مجموعة الاختبار واختر حالة اختبار جديدة.



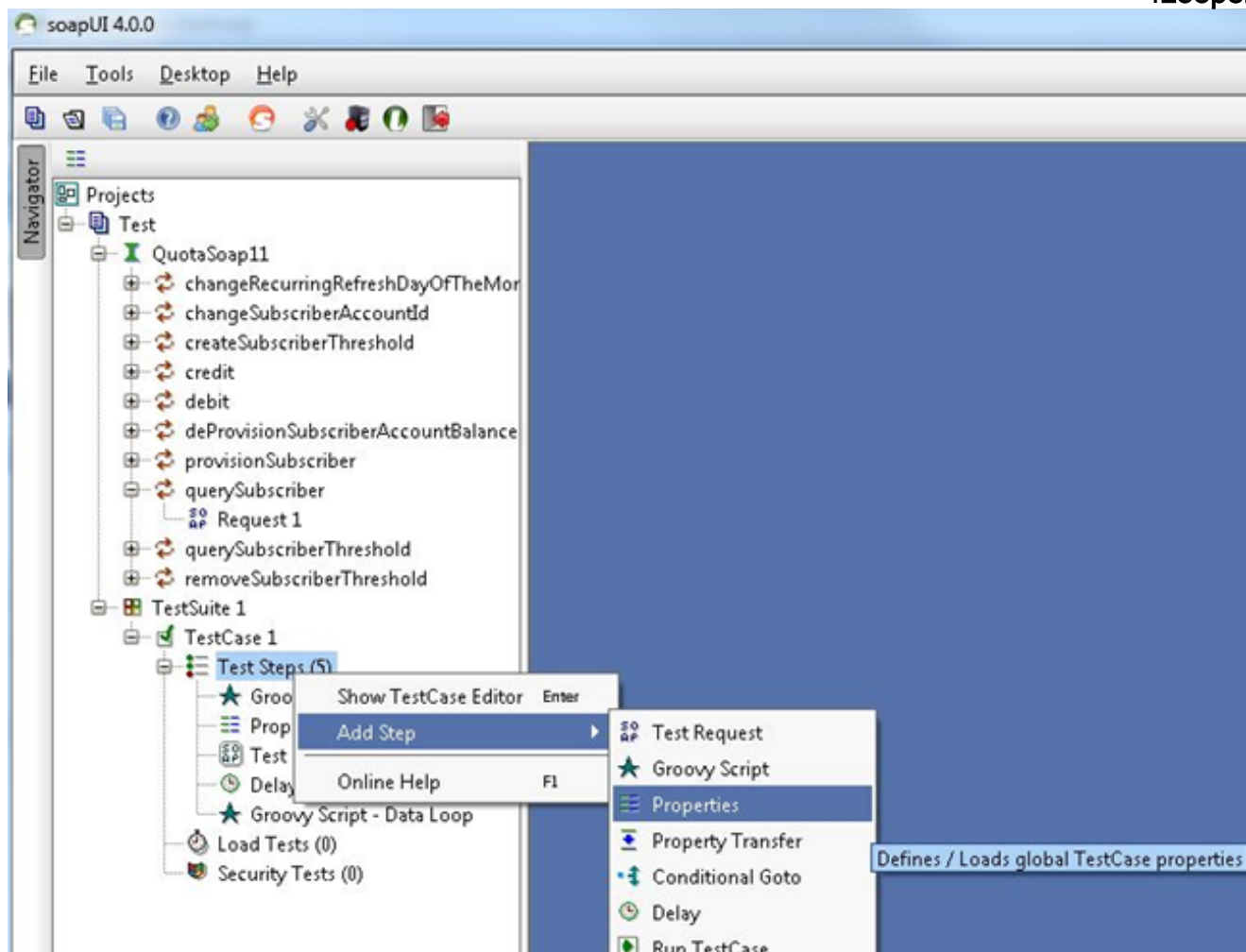
3. انقر بزر الماوس الأيمن على حالة الاختبار واختر إضافة خطوة < نص تنفيذي لإضافة خطوة اختبار برنامج نصي. قم بتسمية مصدر بيانات تقنية المعلومات:



4. في ملف "مصدر البيانات"، قم بلصق هذا الرمز. يقرأ هذا الرمز الملف **C:/subid.txt** الذي يحتوي على معرف المشترك في كل سطر:

```
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.wsd1.testcase.Wsd1TestCaseRunner
}testRunner.testCase, null) propTestStep.setPropertyValue("StopLoop", "F") } else)
{ ("propTestStep.setPropertyValue("StopLoop", "F
```

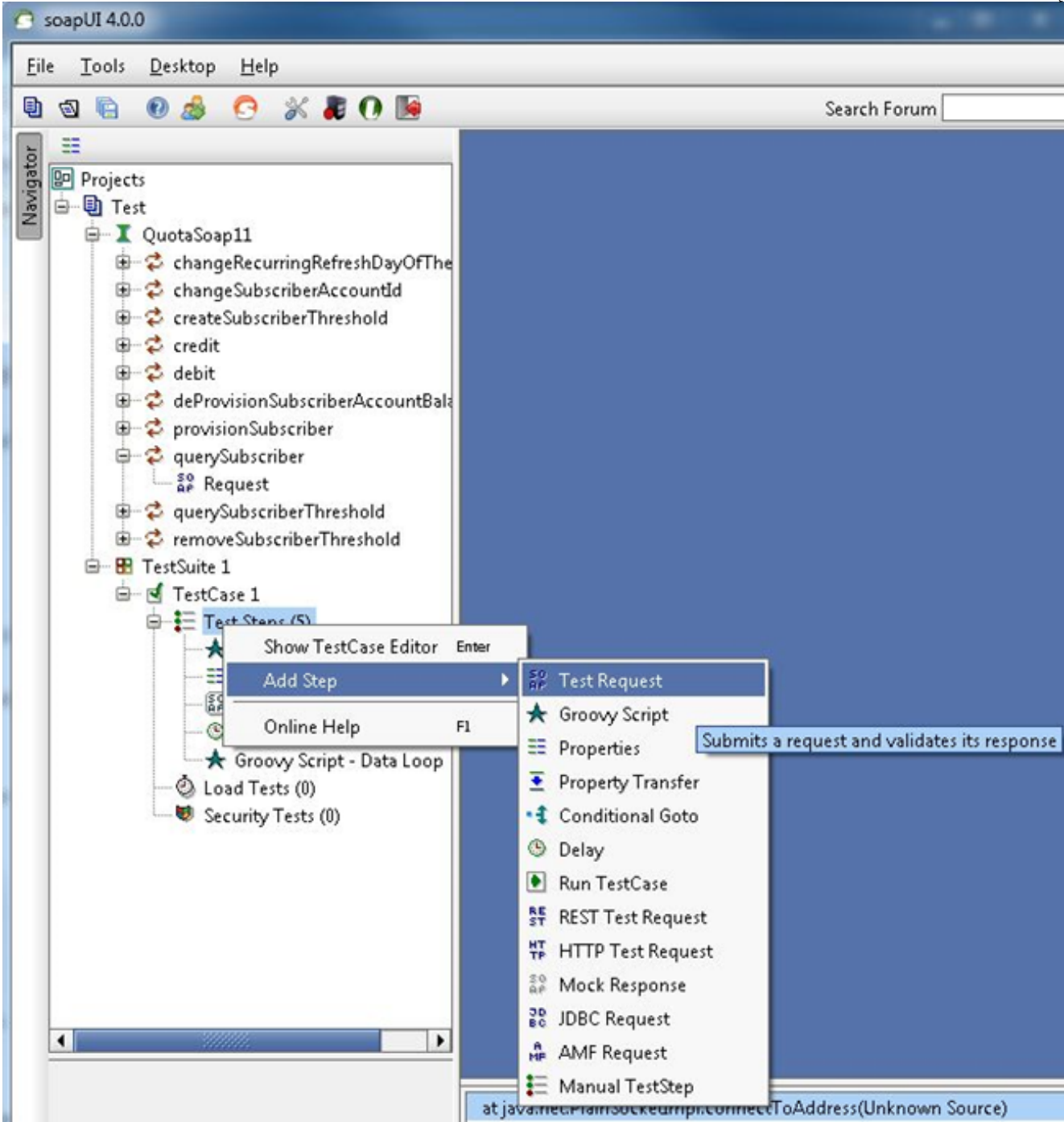
5. انقر بزر الماوس الأيمن فوق خطوة الاختبار واختر **إضافة خطوة** < خصائص لإضافة خطوة اختبار خاصة وتسميتها خاصة -



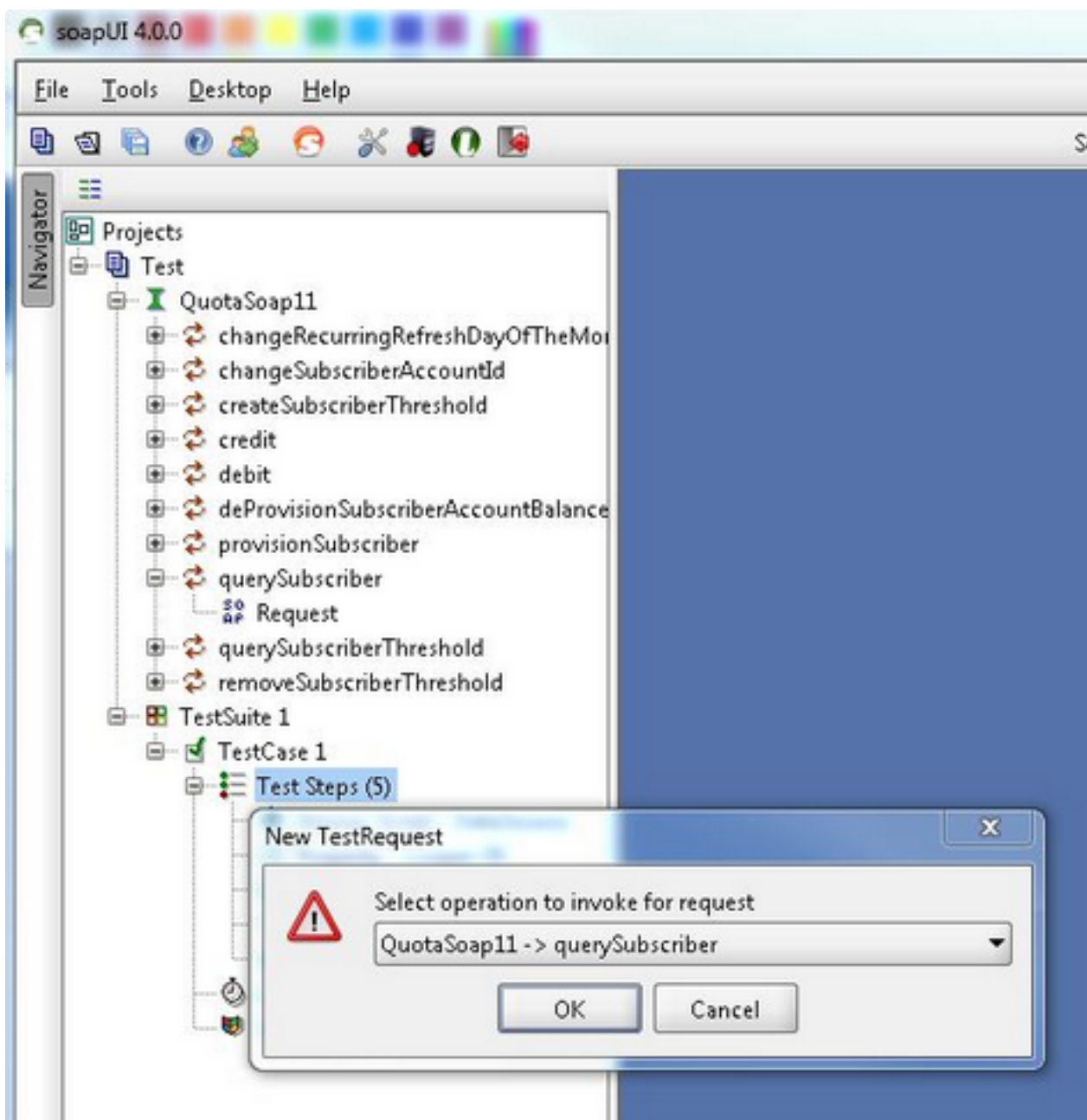
6. أضفت هذا يعرف خاصية من المستعمل خطوة إختبار Looper: المجموع والقيمة - في المثال الذي نضربه، يؤدي هذا إلى الاحتفاظ بمعرف المشترك المقروء من معرفات مشترك الملفعد التالياييقاف الحلقة

Name	Value
Total	
Value	
Count	
Next	
StopLoop	

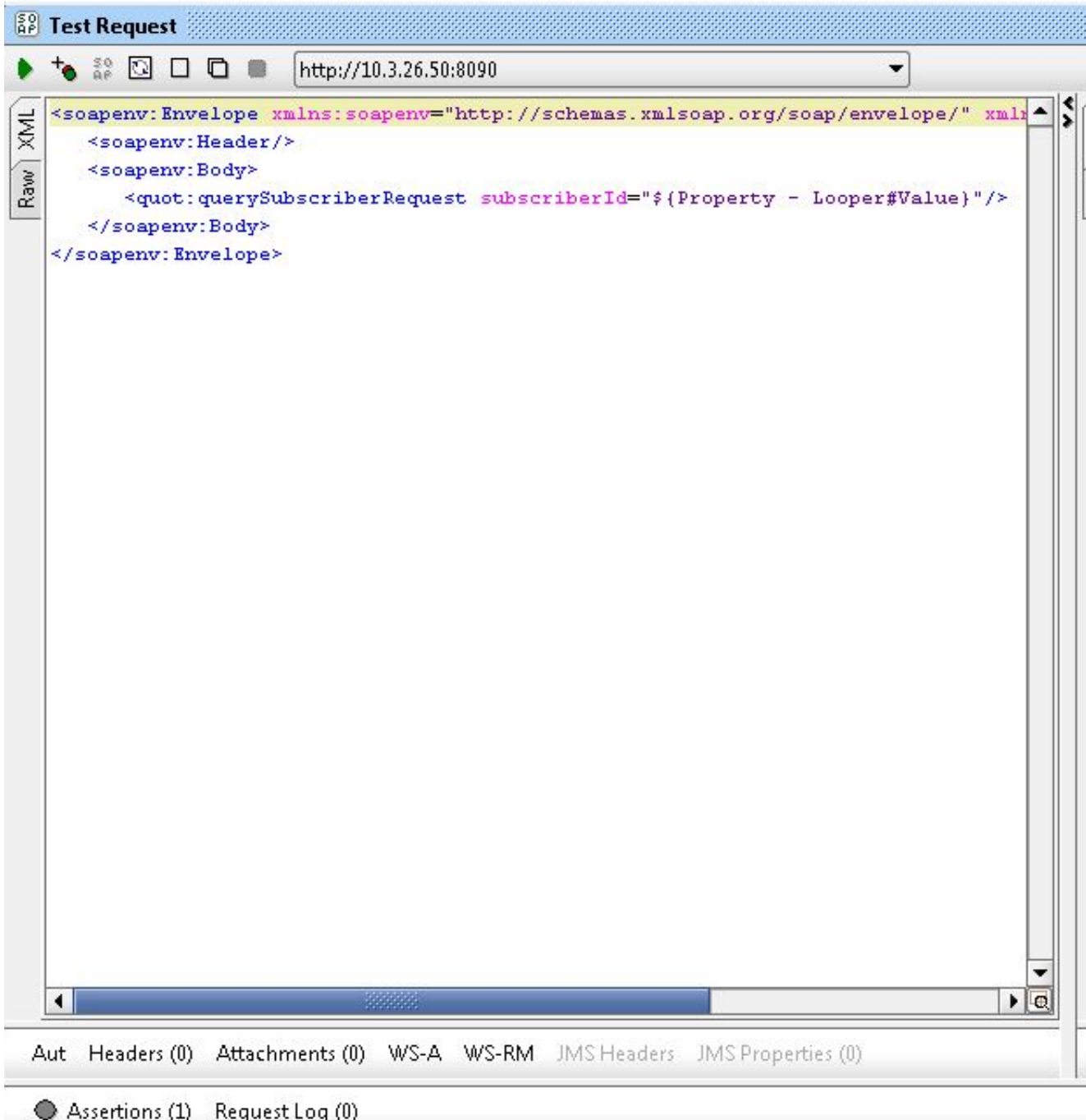
7. انقر بزر الماوس الأيمن فوق خطوة الاختبار واختر إضافة خطوة > طلب الاختبار لإضافة خطوة إختبار إختبار واختر الطلب الذي تريد إستدعائه:



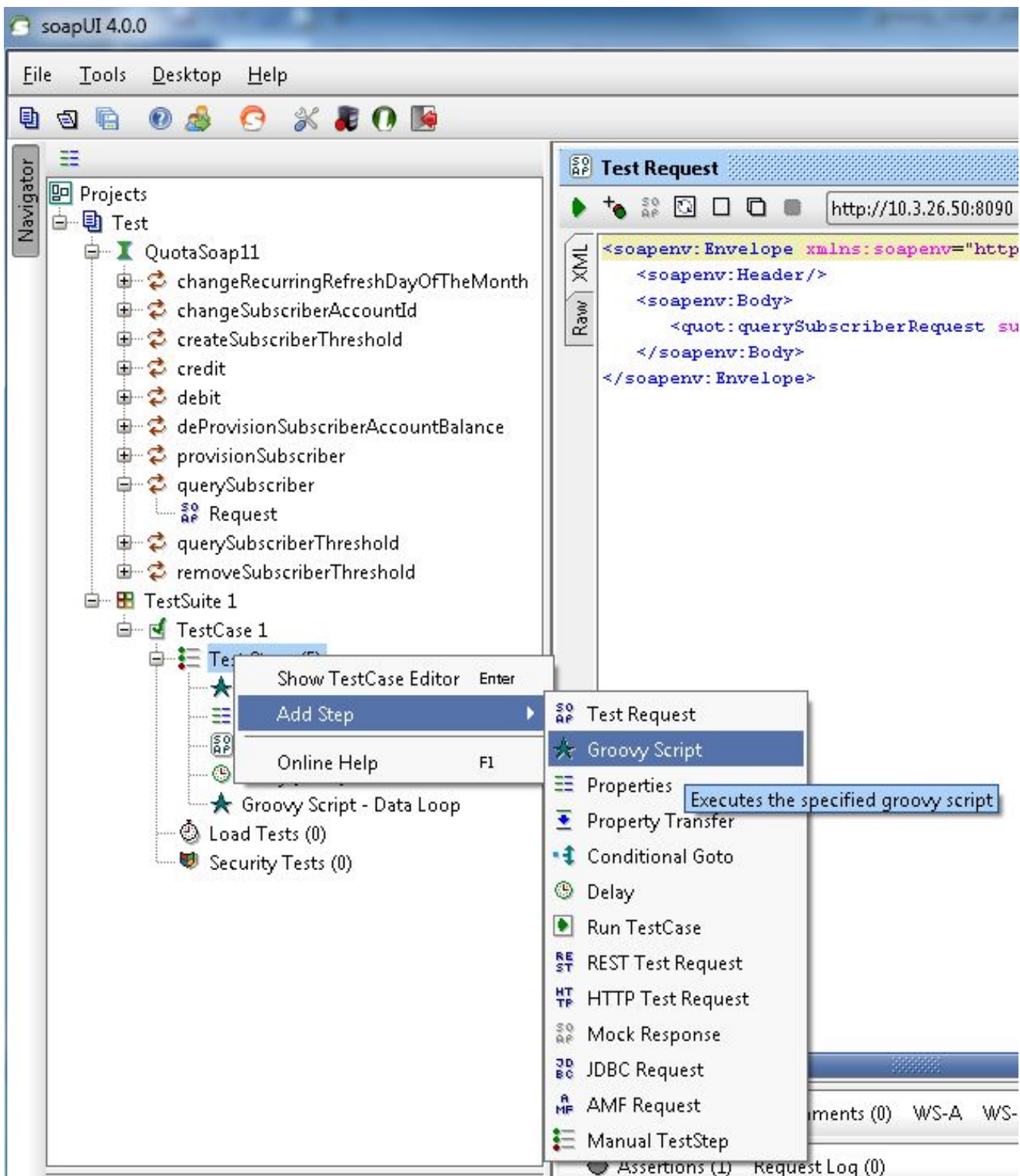
في هذا المثال، يتم استخدام querySubscriberRequest.



8. في الطلب، يستبدل رمز التوسيع قيم الحقل الخاصة بما تستعلم عنه. في هذا المثال، يتم إستبدال ؟ من `Property - Looper#Value}}$` في `querySubscriberRequest` برمز توسيع `?(=SubscriberId`
`::(soap_test_req_expansion_code`



الخاصية - Looper هو اسم TestStep الذي تم إنشاؤه مسبقاً وتحتوي القيمة على معرف المشترك الحالي الذي تمت قراءته من ملف معرفات المشترك.
9. انقر بزر الماوس الأيمن فوق خطوة الاختبار واختر إضافة خطوة < برنامج نصي للتجميع وتسميته حلقة البيانات:



10. الصق هذا الرمز في حلقة بيانات البرنامج النصي 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. في هذا مثال إجراء، 1000 مللي ثانية تأخير بين كل أنشطة أضفت. هذه الخطوة إختيارية. ومع التأخير هناك الآن خمس خطوات تجريبية:

The screenshot displays the JMeter interface. On the left, the 'Navigator' pane shows a tree structure under 'Test' with a sub-tree '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 main window shows 'TestCase 1' with a toolbar and a 'TestSteps' list. Below the list are tabs for 'Description', 'Properties', 'Setup Script', and 'TearDown Scrip'. At the bottom, the 'Test Properties' table is visible.

Property	Value
Name	TestCase 1

The console output at the bottom shows a stack trace error:

```

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

```

12. انقر فوق زر تشغيل الأخضر لتشغيل خطوات الاختبار الخمس في نافذة TestCase.

ةمچرتل هذه لوج

ةللأل تاي نقتل نمة ومة مادختساب دن تسمل اذة Cisco تمةرت
ملاعلاء انء مء مء نمة دختسمل معد و تمة مء دقتل ةر شبل او
امك ةق قء نوك ت نل ةللأل ةمچرت لصف أن ةظحال مء ءرء. ةصاأل مء تءل ب
Cisco ةللخت. فرتمة مچرت مء دقء ةللأل ةل فارتحال ةمچرتل عم لاعل او
ىل إأمءءاد ءوچرلاب ةصوء و تامةرتل هذه ةقء نء اهءل وئس م Cisco
Systems (رفوتم طبارل) ةلصلأل ةزءل ءن إل دن تسمل