

使用Groovy指令碼自動化API

目錄

[簡介](#)

[建立soapUI專案](#)

[建立soapUI API請求](#)

[建立soapUI測試用例](#)

簡介

本文檔介紹如何建立soapUI應用程式程式設計師介面(API)請求，以及如何建立soapUI測試用例，該測試用例循環執行測試步驟，以自動向Quantum策略套件(QPS)發出API請求。

本文中的示例soapUI測試用例實施了一個測試步驟，該測試步驟讀取了一個使用者ID的檔案，然後建立並傳送一個querySubscriberRequest到QPS。

建立soapUI專案

開始此過程之前，請在案頭上安裝soapUI應用程式。您可以從www.soapui.org下載soapUI安裝執行檔。

建立API請求或測試用例之前，必須先建立soapUI專案。建立專案需要Web服務描述語言(WSDL)檔案和XML架構描述(XSD)檔案。WSDL指定支援的API。從負載平衡器(LB)運行這些命令時，通常可以從QPS獲取WSDL和XSD：

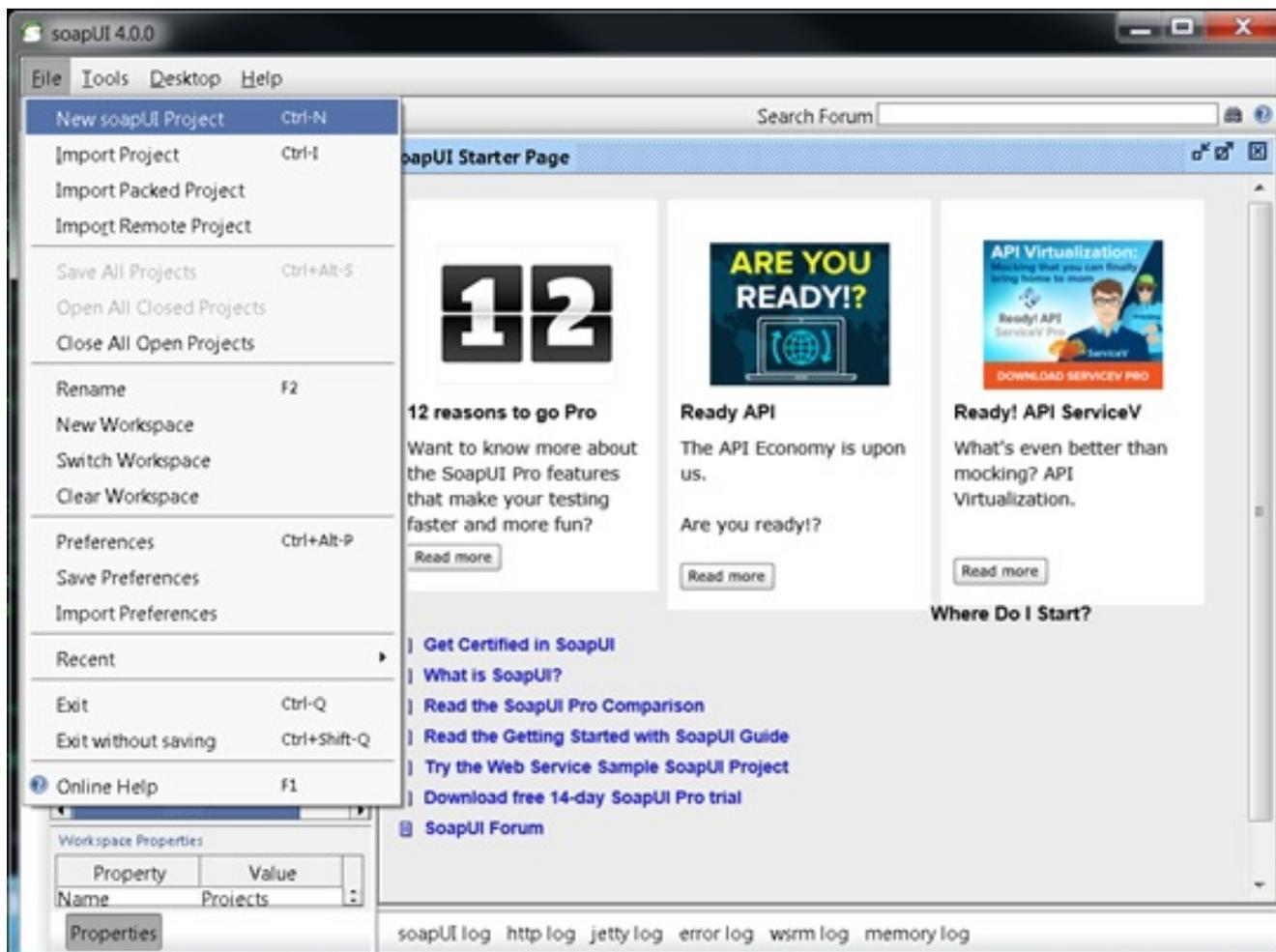
- `wget http://lbvip01:8080/ua/wsd/UnifiedApi.wsd`
- `wget http://lbvip01:8080/ua/wsd/UnifiedApi.xsd`

將WSDL和XSD儲存在案頭上計畫運行soapUI應用程式的同一目錄中。

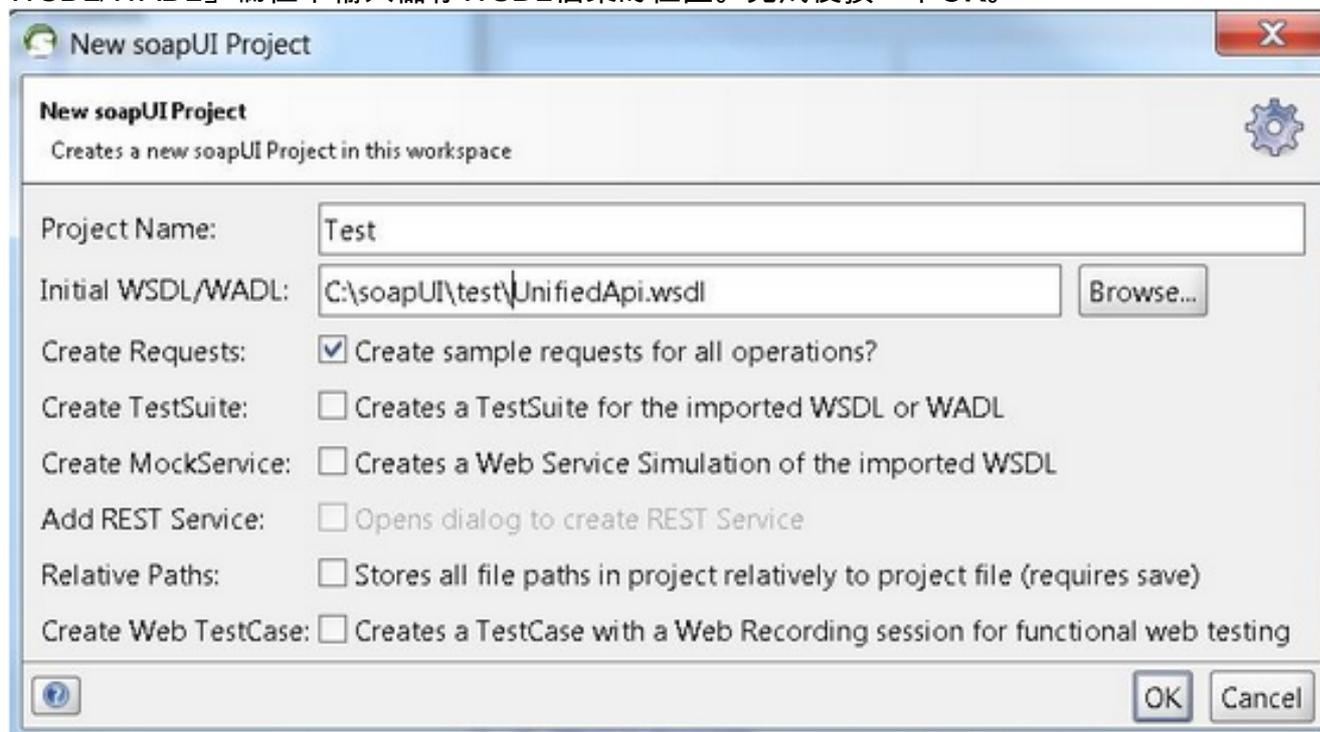
完成以下步驟以建立soapUI專案：

1. 從soapUI視窗中選擇「**檔案**」>「**新建soapUI專案**」

：



2. 在「新建soapUI專案」視窗的「專案名稱」欄位中輸入專案的名稱，並在「初始WSDL/WADL」欄位中輸入儲存WSDL檔案的位置。完成後按一下OK。

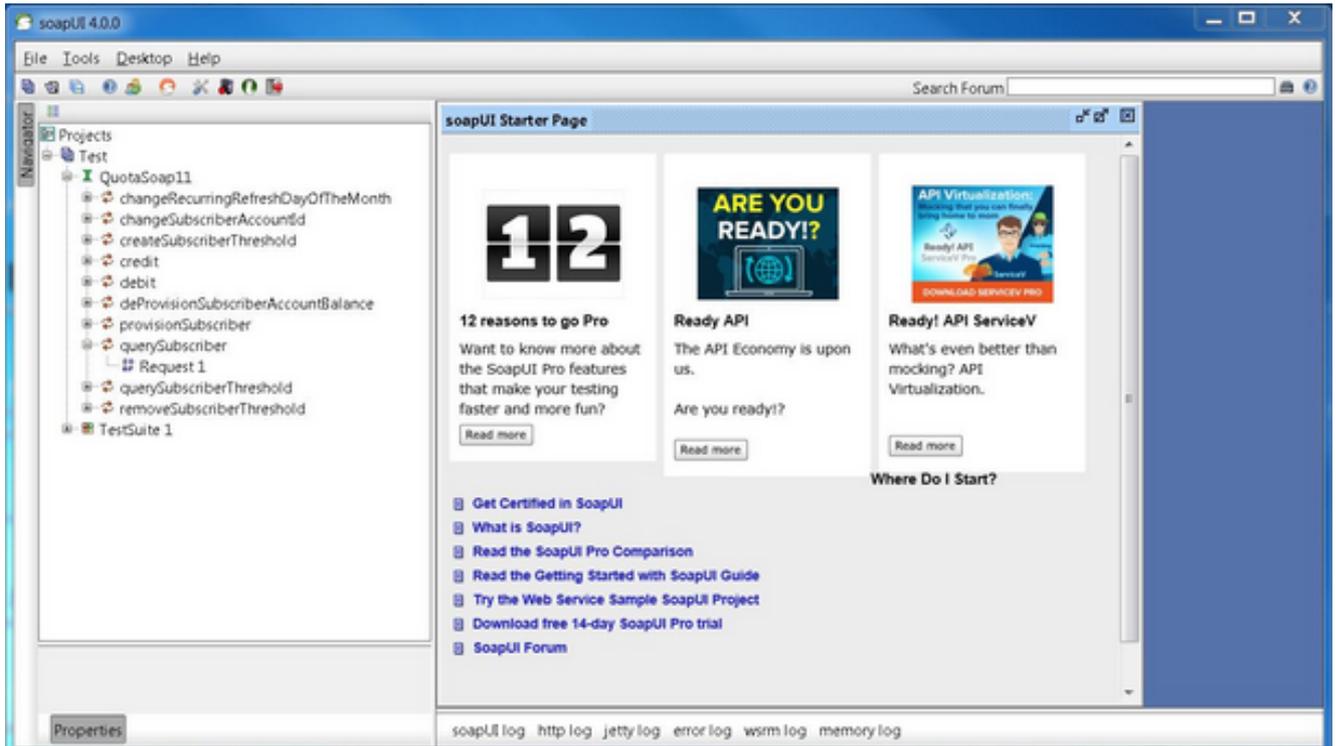


建立soapUI API請求

完成以下步驟以建立soapUI API請求：

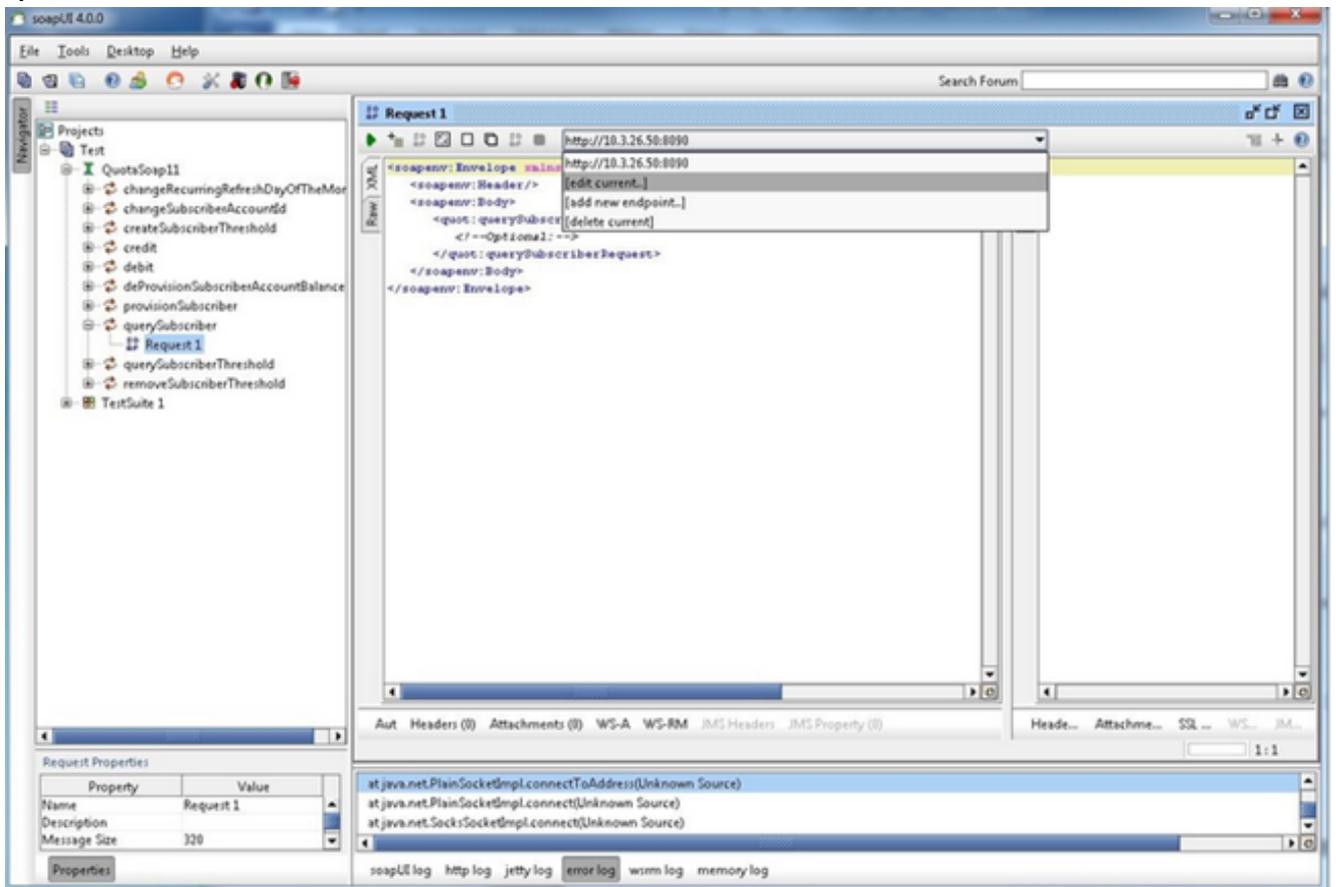
1. 展開您建立的soapUI專案以檢視API。您還可以展開其中一個API以檢視請求。在本示例中，querySubscriberRequest已展開

:



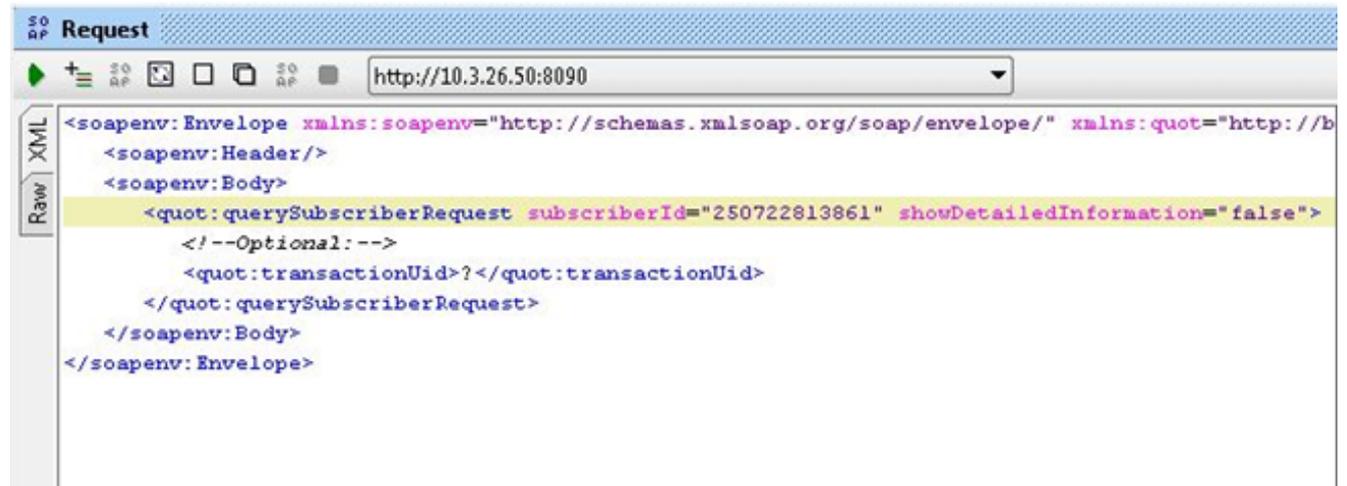
2. 開啟請求，以便使用構成查詢的XML檢視請求視窗。在「請求」視窗中，將http:// IP地址編輯為IP地址和埠。這通常是您要傳送請求的Ibvip01 IP地址和埠，如以下示例所示

:



3. 使用要在請求中傳送的資料修改XML中的欄位。在本示例中，請求是querySubscriberRequest。修改要查詢的訂閱伺服器的訂閱伺服器ID，並將showDetailedInformation設定為

false:



4. 按一下「請求」視窗頂部的**運行綠色按鈕**以運行查詢。

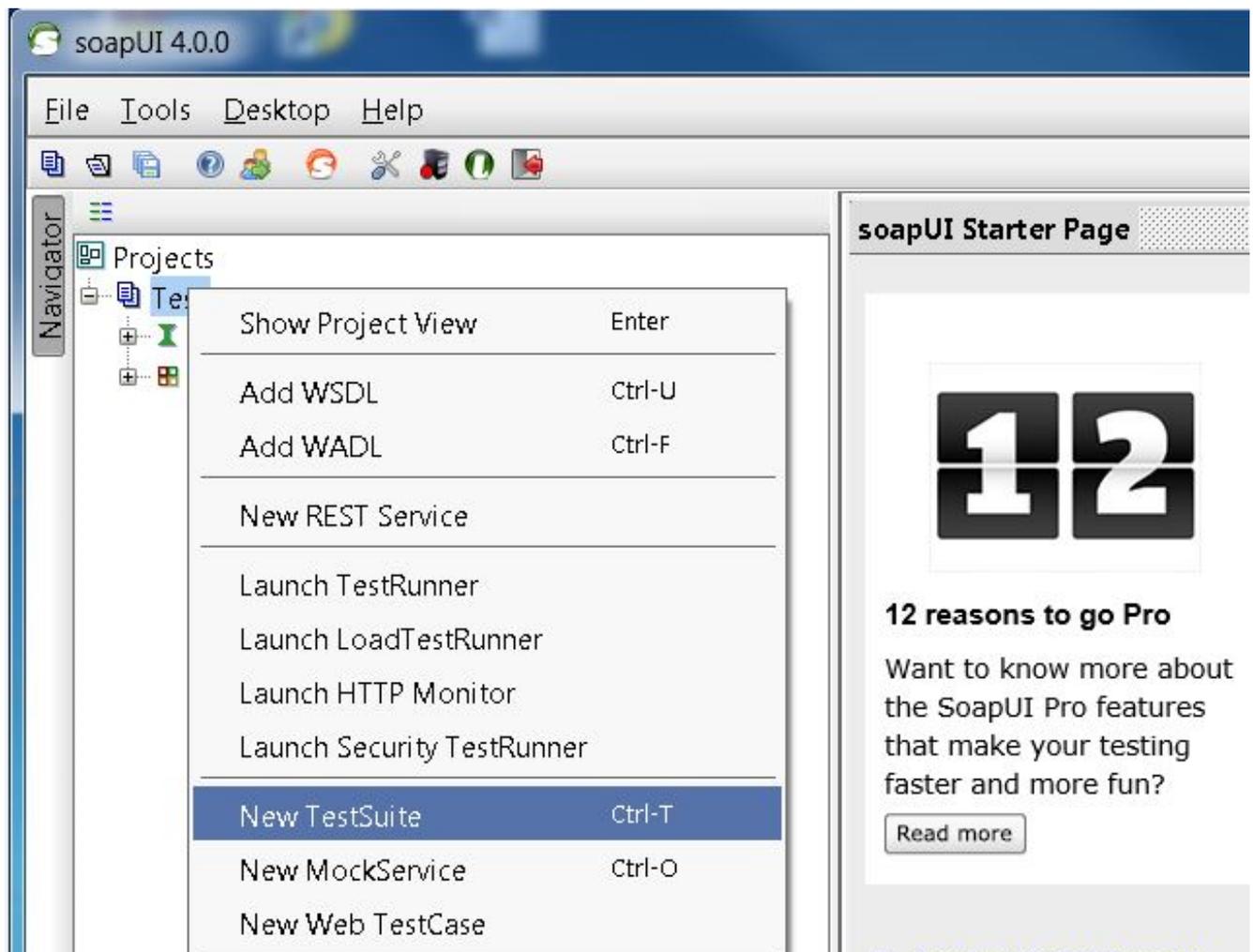
建立soapUI測試用例

以下過程說明如何建立測試套件，該套件可在API傳送到QPS時實現自動化。

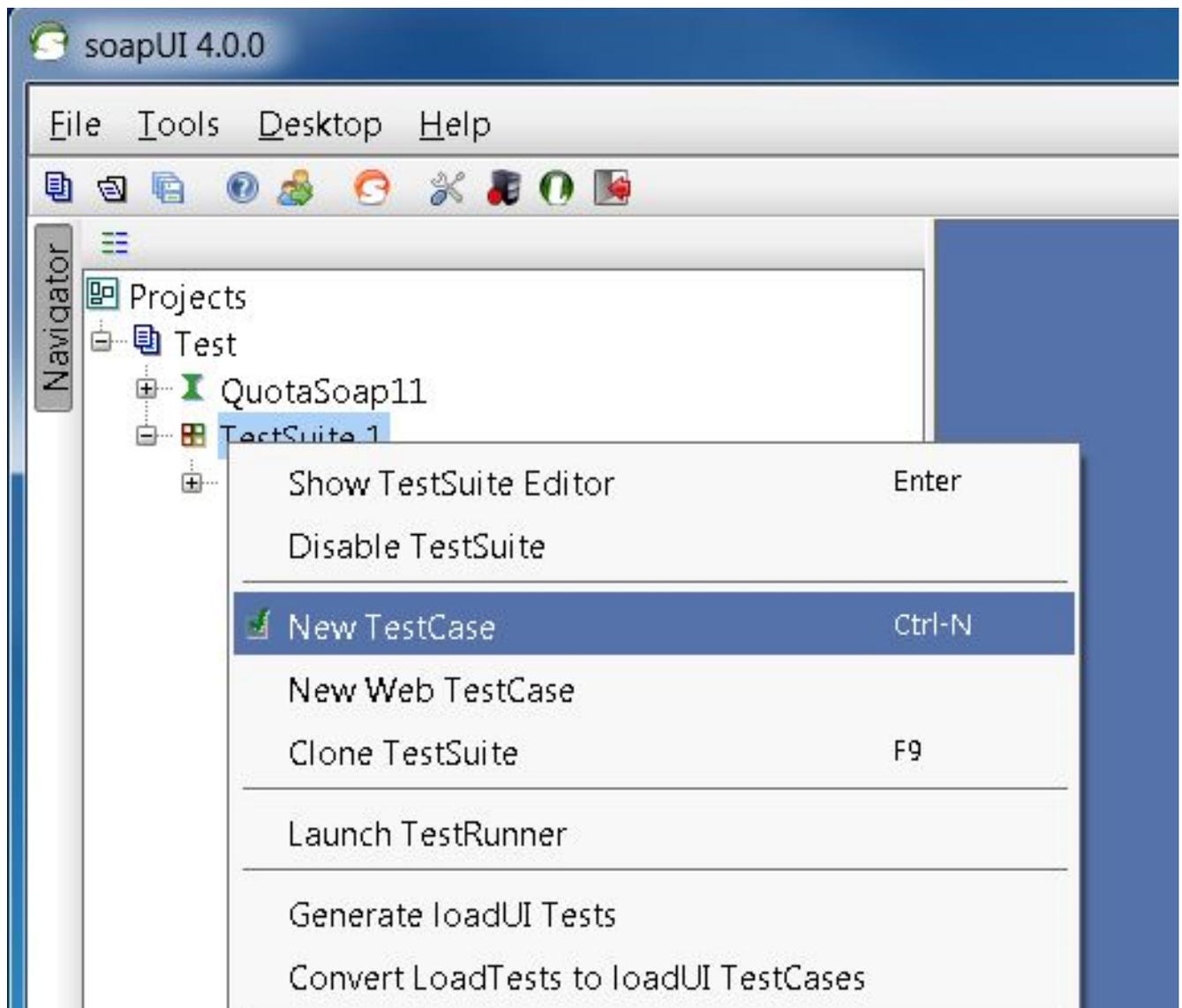
在此示例過程中，測試套件循環訪問一組使用者ID，然後使用它傳送到QPS的querySubscriberRequest中的這些使用者ID。每個訂戶ID清單位於名為**subid.txt**的文本檔案中的單行上。

完成以下步驟以建立測試套件：

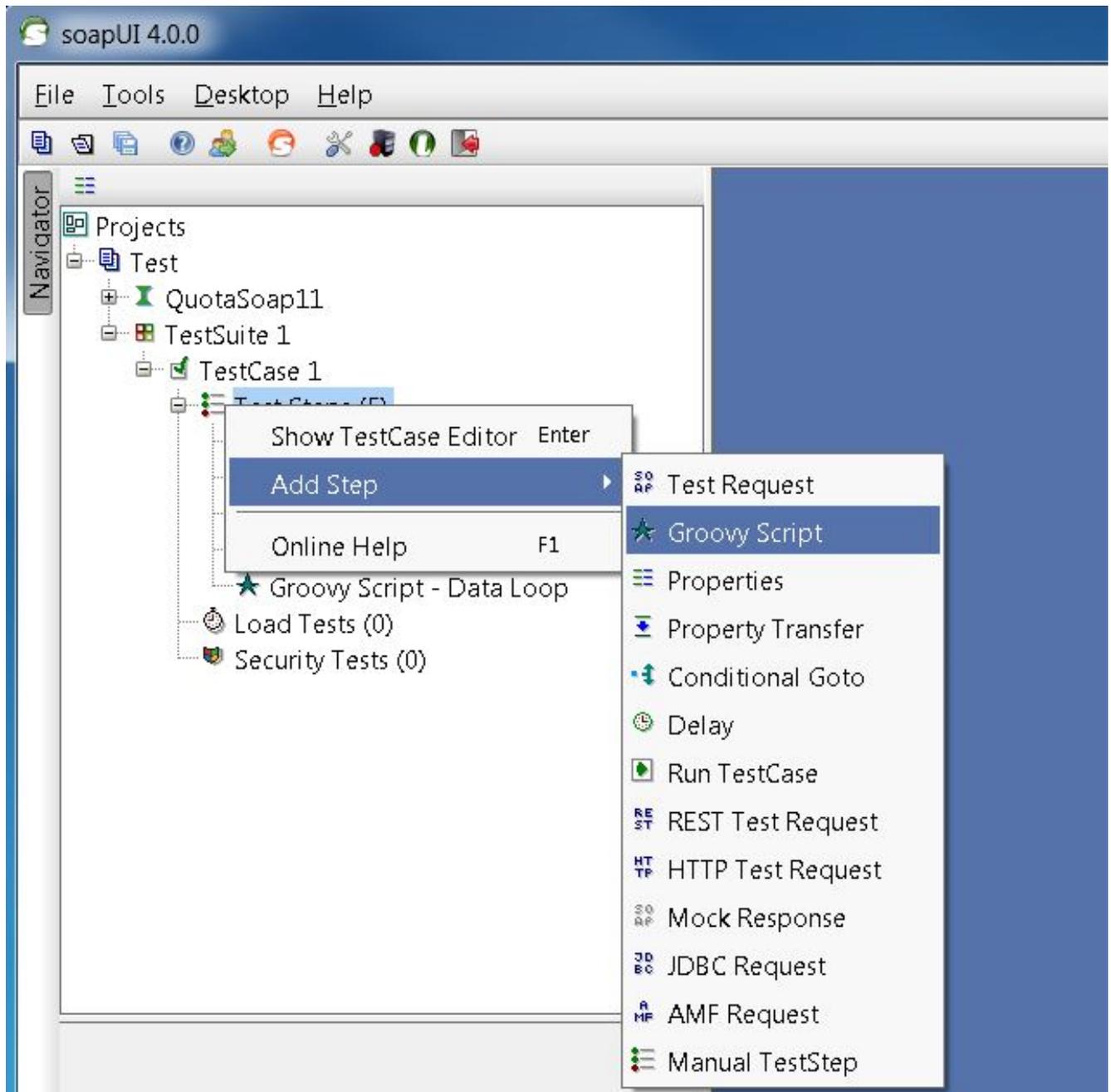
1. 在您建立的soapUI專案中，建立一個新的測試套件。按一下右鍵soapUI並選擇**New TestSuite**。



2. 按一下右鍵測試套件並選擇**New TestCase**。



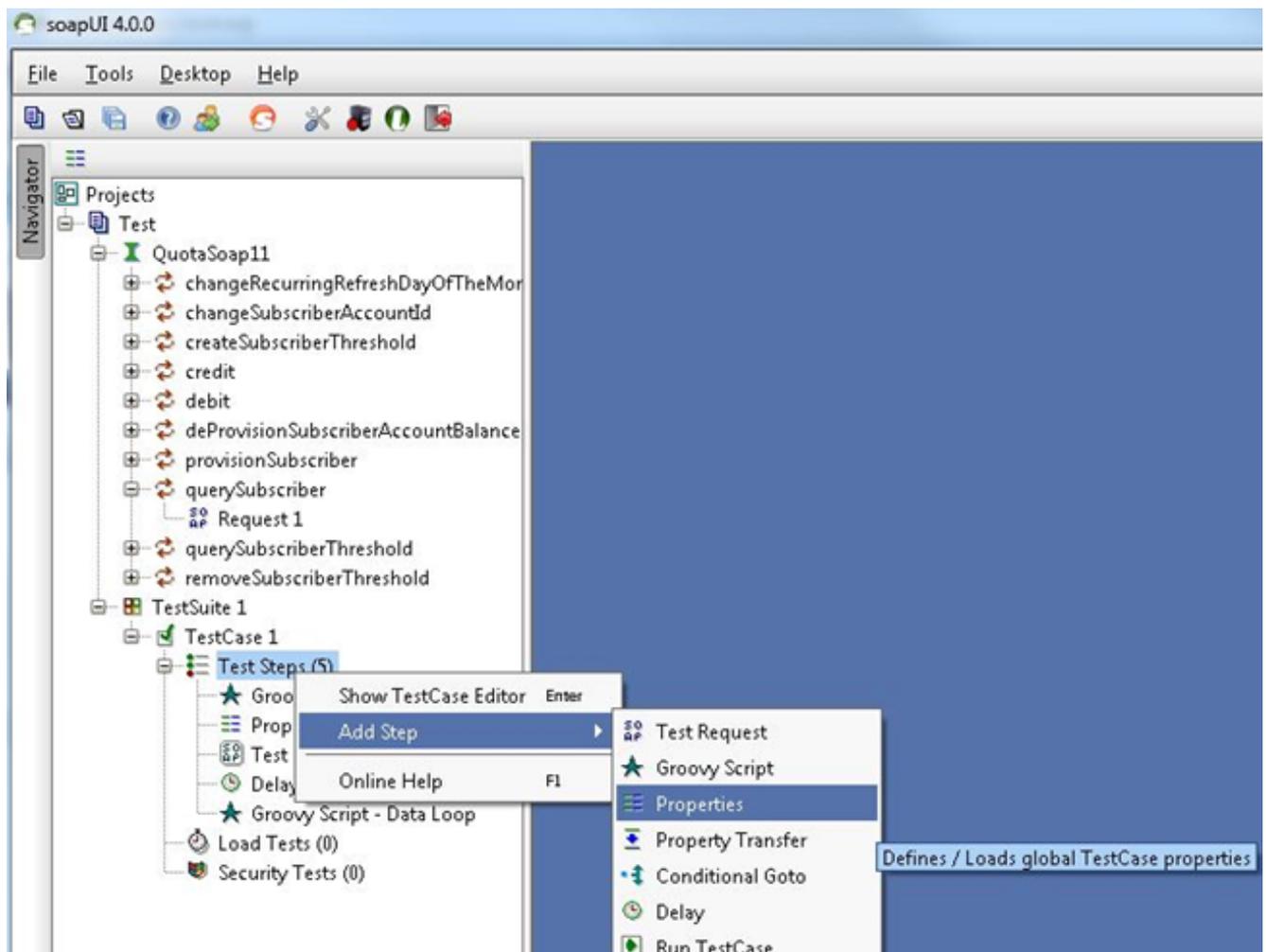
3. 按一下右鍵測試用例，然後選擇Add Step > Groovy Script以新增Groovy Script測試步驟。將其命名為資料來源：
：



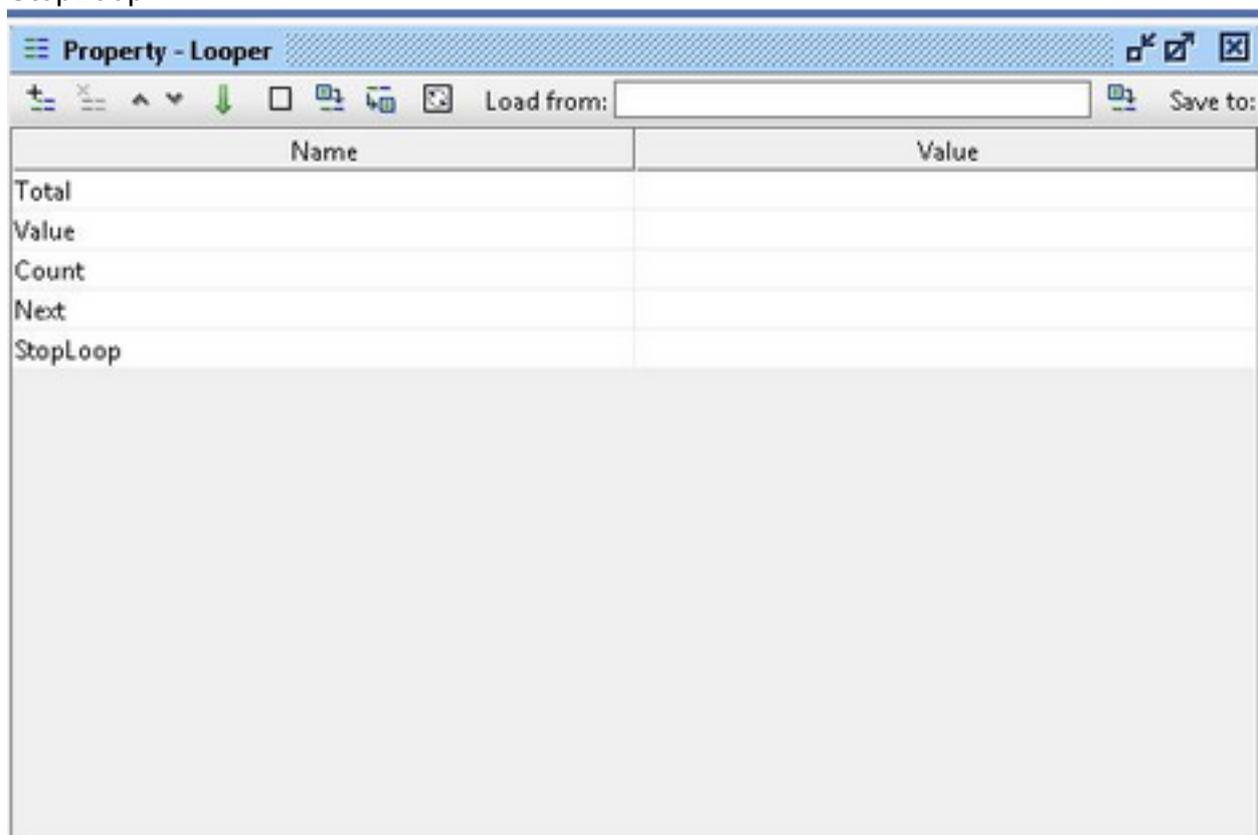
4. 在資料原始檔中貼上此代碼。此代碼讀取檔案C:/subid.txt，該檔案包含每條線路上的訂戶ID:

```
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. 按一下右鍵測試步驟，然後選擇Add Step > Properties以新增屬性測試步驟並將其命名為Property - Looper。



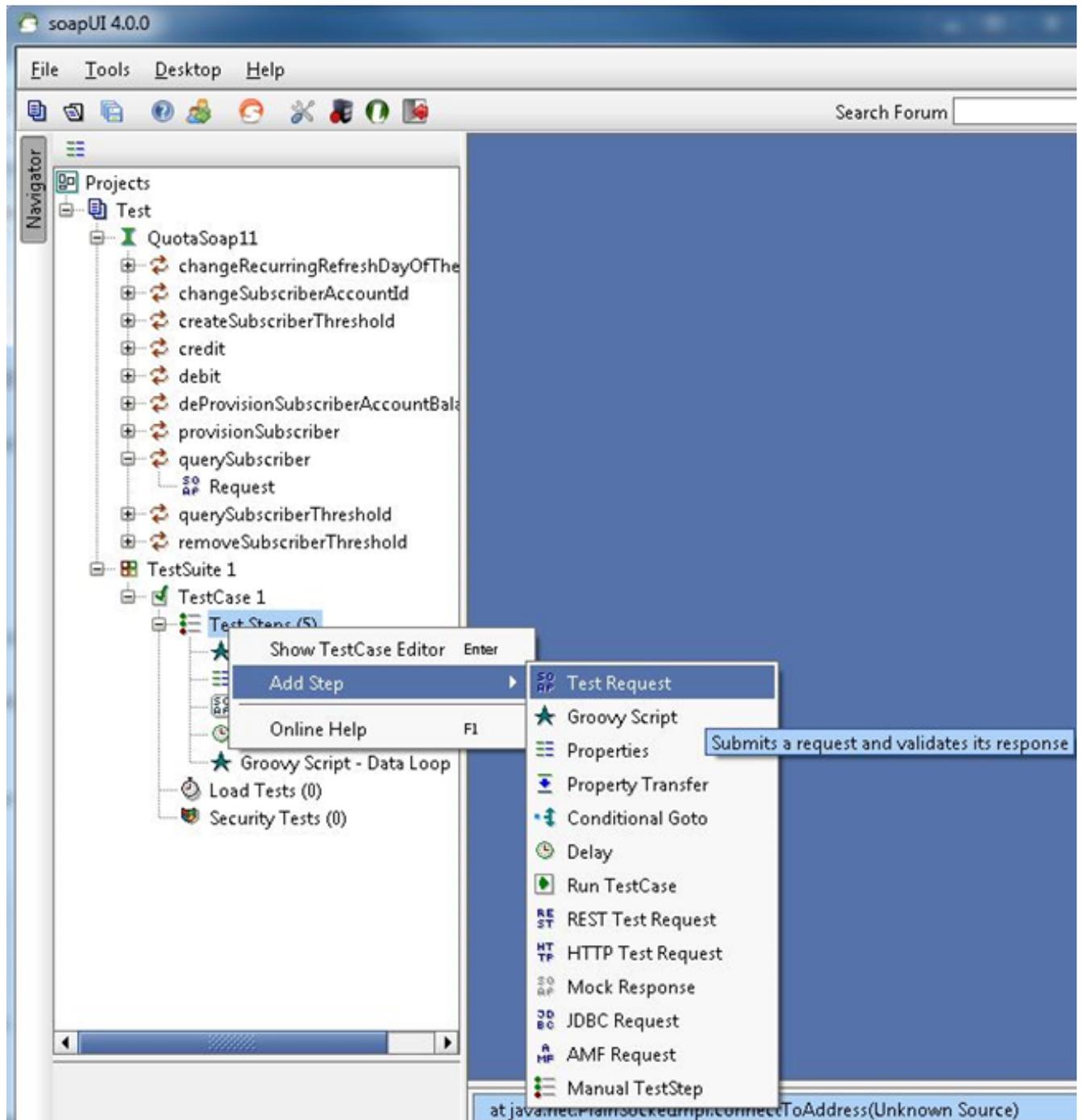
6. 新增以下使用者定義的Looper測試步驟屬性：總計Value — 在我們的示例中，儲存從檔案訂戶ID讀取的訂戶ID計數下一頁
StopLoop



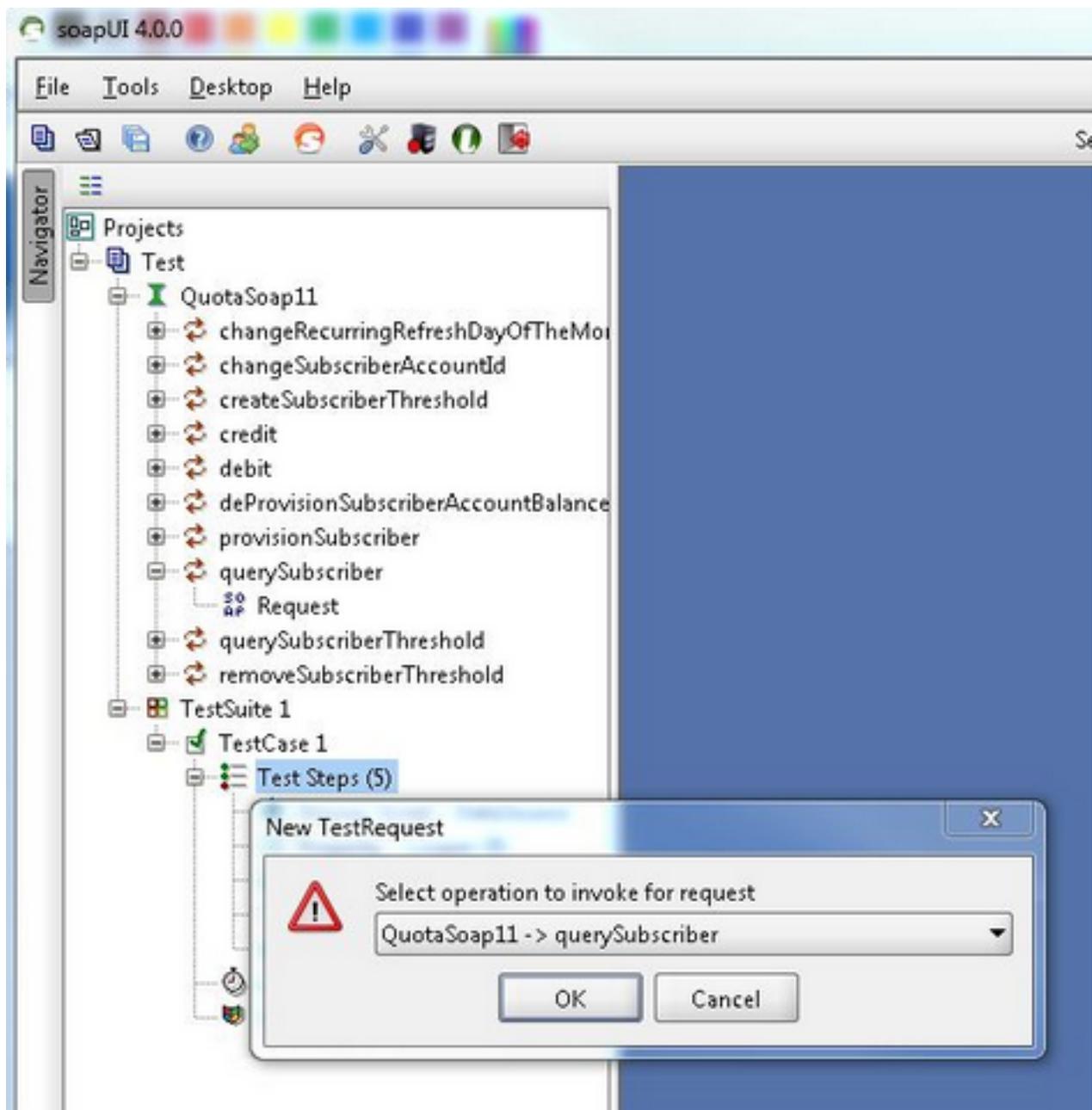
7. 按一下右鍵測試步驟，然後選擇Add Step > TestRequest以新增測試請求測試步驟，並選擇要

呼叫的請求

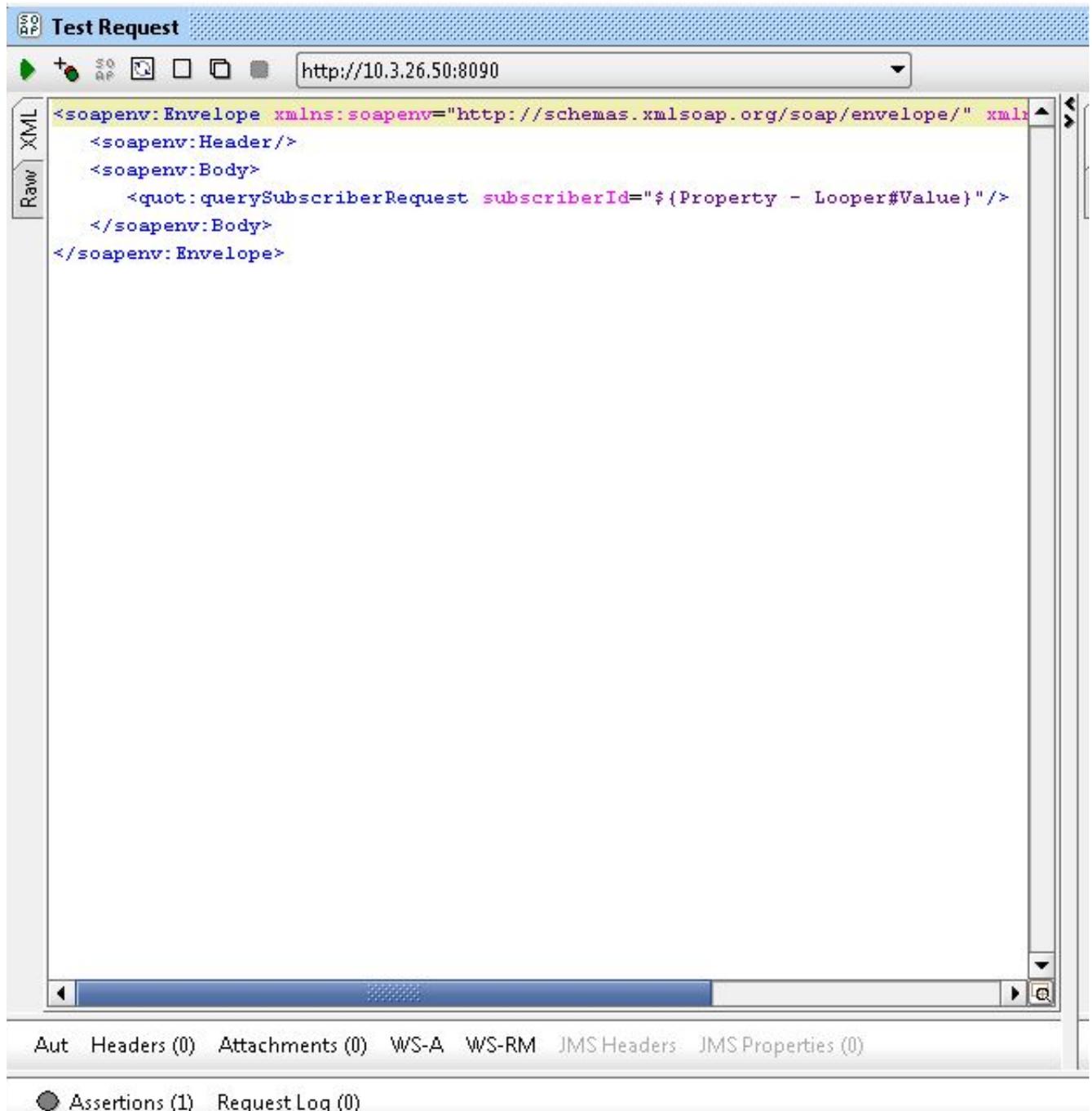
:



在此示例中，使用querySubscriberRequest。

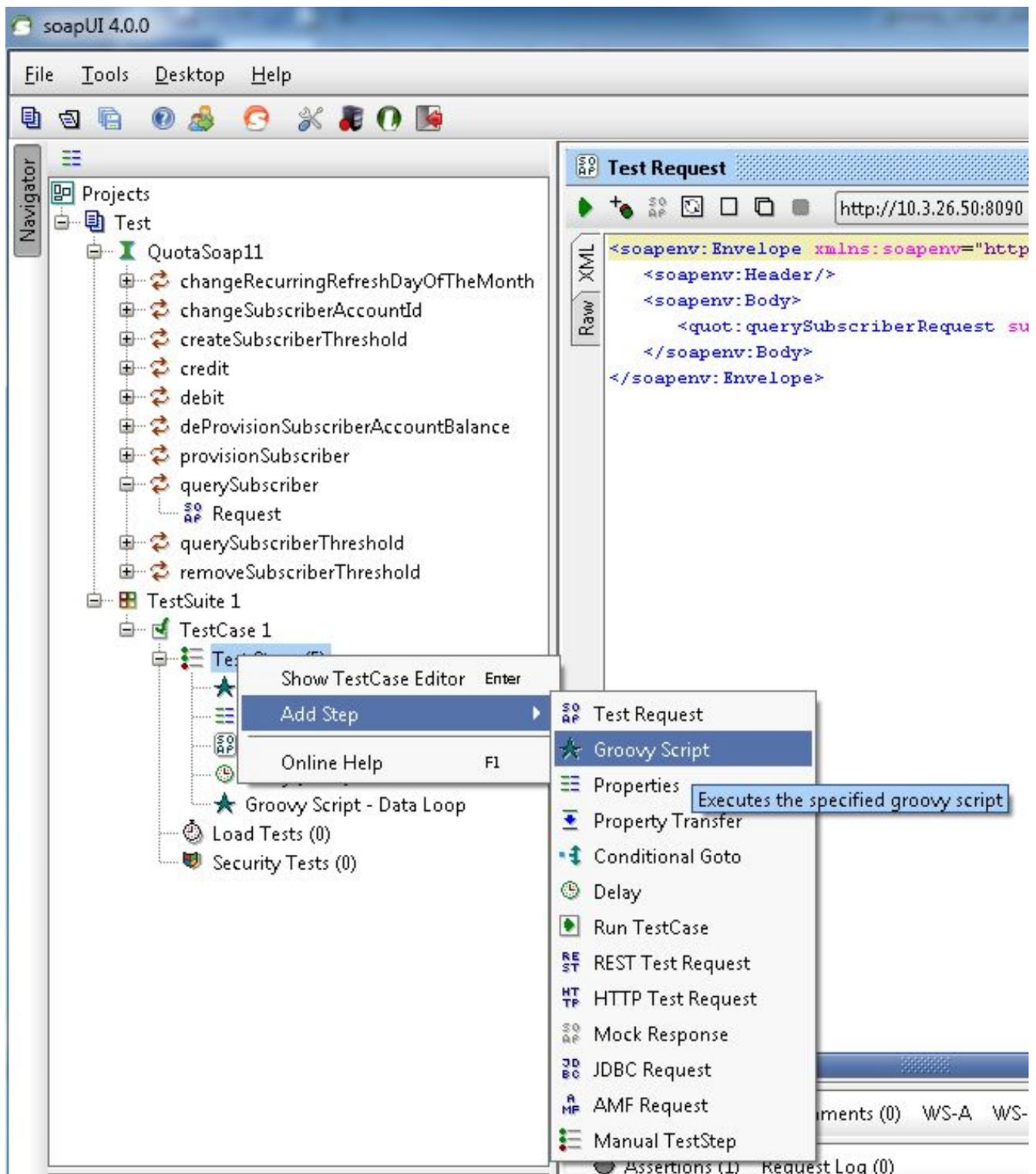


8. 在請求中，擴展代碼將替換所查詢內容的欄位值。在本例中?的SubscriberId=?在 querySubscriberRequest中替換為擴展代碼`$(Property - Looper#Value)(soap_test_req_expansion_code)`:



Property - Looper是以前建立的屬性TestStep的名稱，Value儲存從使用者ID的檔案讀取的當前使用者ID。

9. 按一下右鍵測試步驟，然後選擇Add Step > Groovy Script，並將其命名為Data Loop:



10. 將此代碼貼上到Groovy指令碼資料循環中：

```

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. 在此範例過程中，會在每個回圈之間增加1000 ms延遲。此步驟是可選的。延遲現在有五個測試步驟

:

The screenshot displays a testing tool interface with a Navigator on the left and a TestSteps panel on the right. The Navigator shows a tree structure under 'Test' with a 'QuotaSoap11' folder containing various test steps like 'changeRecurringRefreshDayOfTheMonth', 'changeSubscriberAccountId', etc. Below it is '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 TestSteps panel on the right shows the same five steps. Below the TestSteps panel is a 'Description Properties Setup Script TearDown Scrip' section. At the bottom, there is a 'TestCase Properties' and 'Test Properties' section with a table.

Property	Value
Name	TestCase 1

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

12. 按一下綠色的運行按鈕以在「測試用例」視窗中運行五個測試步驟。