使用Groovy脚本自动执行API

目录

简介

本文档介绍如何创建soapUI应用程序程序员接口(API)请求，以及如何创建通过测试步骤循环的soapUI测试用例，该测试步骤可自动向Quantum Policy Suite(QPS)发出API请求。

本文中的示例soapUI测试案例实现测试步骤，该测试步骤读取用户ID的文件，然后创建查询SubscriberRequst并将其发送到QPS。

创建soapUI项目

在开始此过程之前，请在桌面上安装soapUI应用。可以从www.soapui.org下载soapUI安装可执行文件。

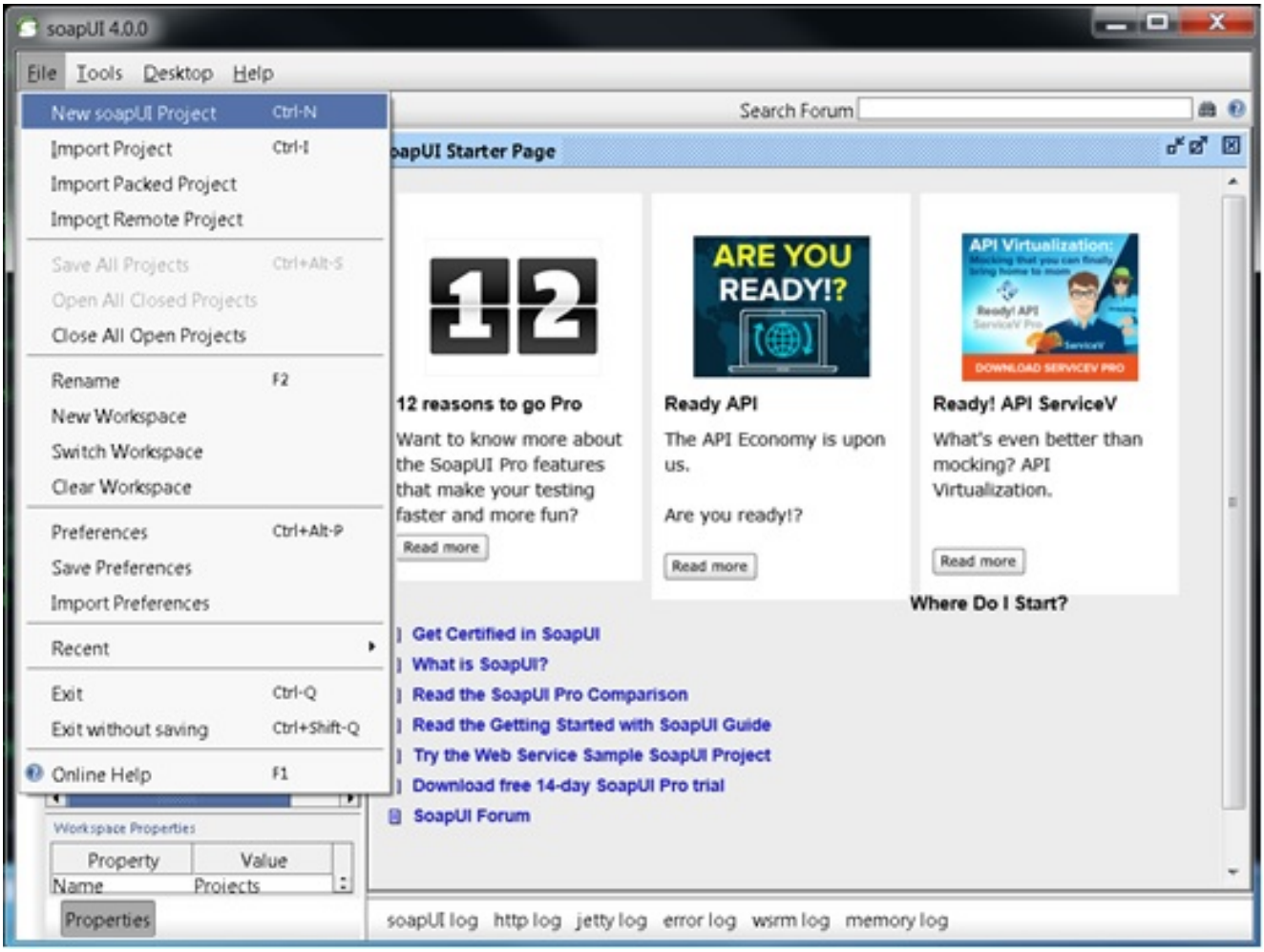
在创建API请求或测试用例之前，必须先创建soapUI项目。要创建项目，需要Web服务描述语言(WSDL)文件和XML架构描述(XSD)文件。WSDL指定支持的API。从负载均衡器(LB)运行以下命令时，通常可以从QPS获取WSDL和XSD:

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

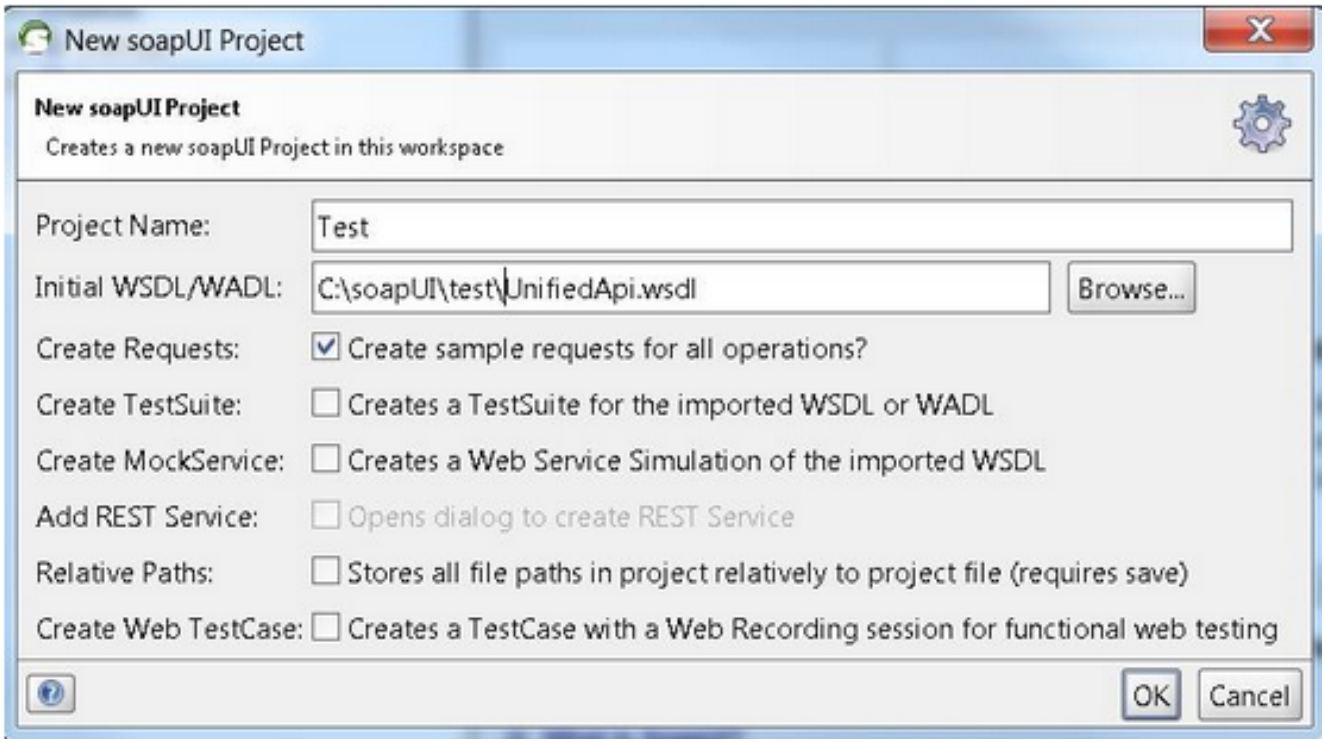
将WSDL和XSD存储在计划运行soapUI应用程序的桌面上的同一目录中。

要创建soapUI项目，请完成以下步骤：

1. 从soapUI**窗口中选择“文件”>“**新建soapUI项目”：



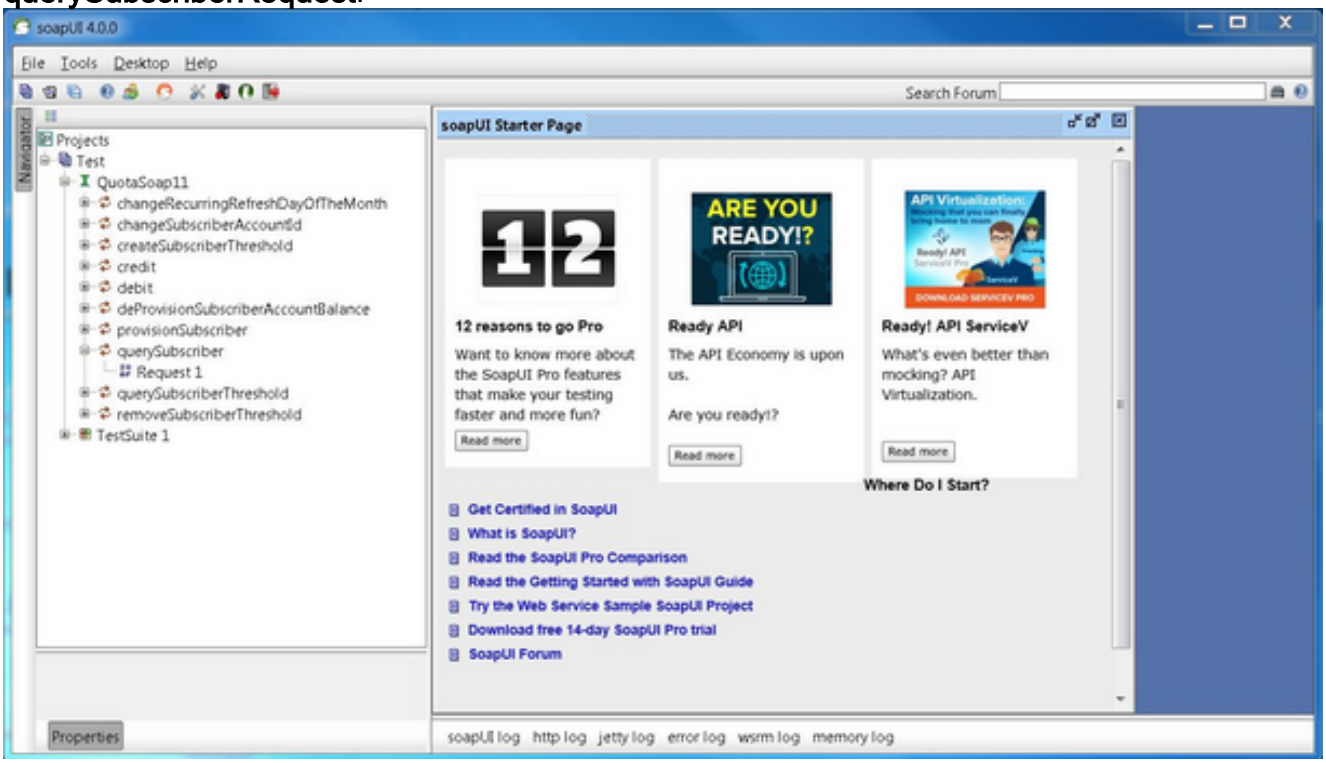
2. 在“新建soapUI项目”窗口中，在“项目名称”字段中输入项目名称，并在“初始WSDL/WADL”字段中输入WSDL文件存储的位置。完成后单击 **OK**。



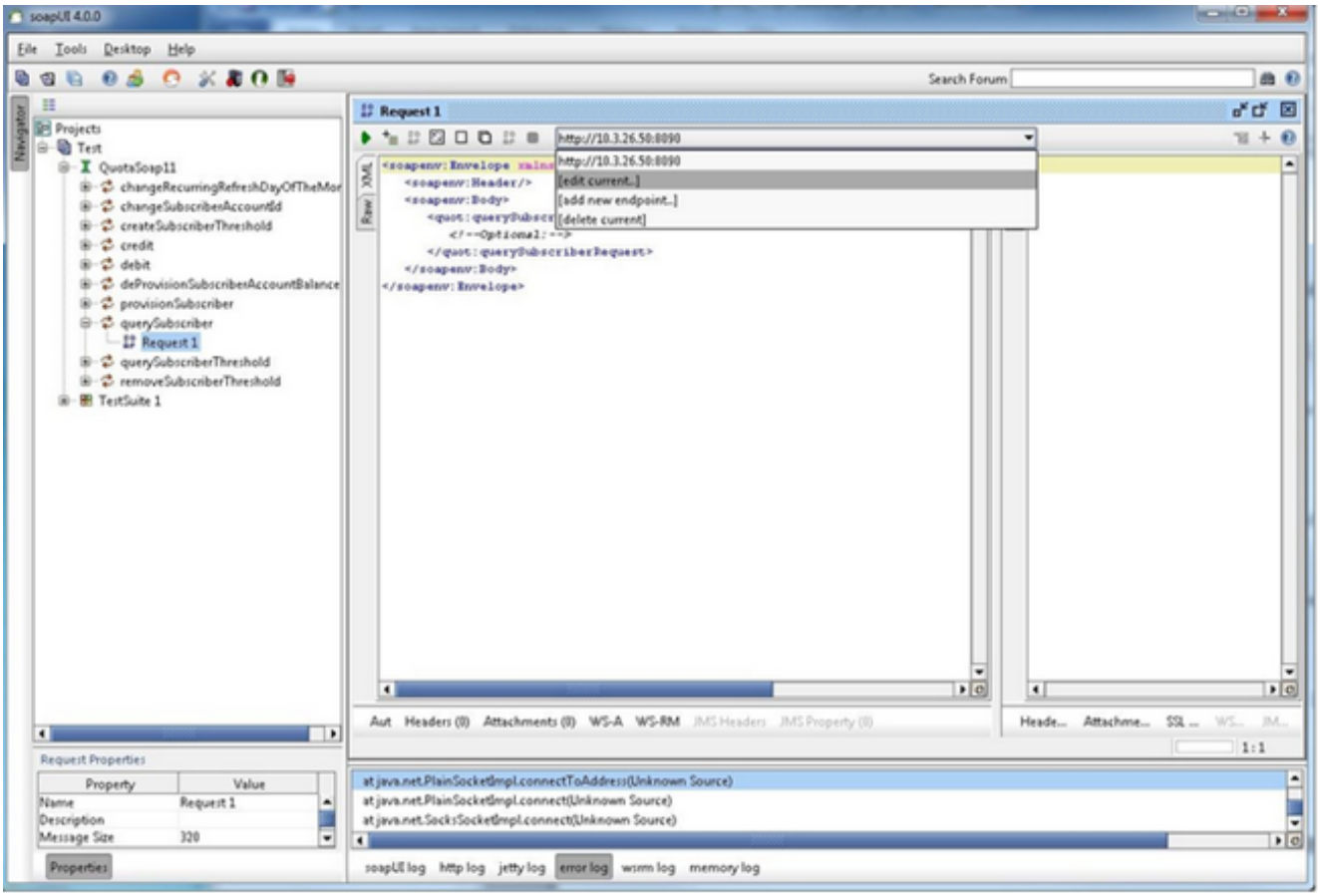
创建soapUI API请求

要创建soapUI API请求，请完成以下步骤：

1. 展开您为查看API而创建的soapUI项目。您还可以展开其中一个API以查看请求。在本例中，展开了
   querySubscriberRequest:



2. 打开请求，以便查看包含XML的请求窗口，该XML用于形成查询。在“请求”窗口中，将http://IP地址编辑为IP地址和端口。这通常是要发送请求的lbvip01 IP地址和端口，如以下示例所示：



3. 使用要在请求中发送的数据修改XML中的字段。在本例中，请求是querySubscriberRequest。修改要查询的订户的订户ID，并将showDetailedInformation设置为
   false:

4. 单击“请求”窗口顶部的绿色“运行”按钮以运行查询。

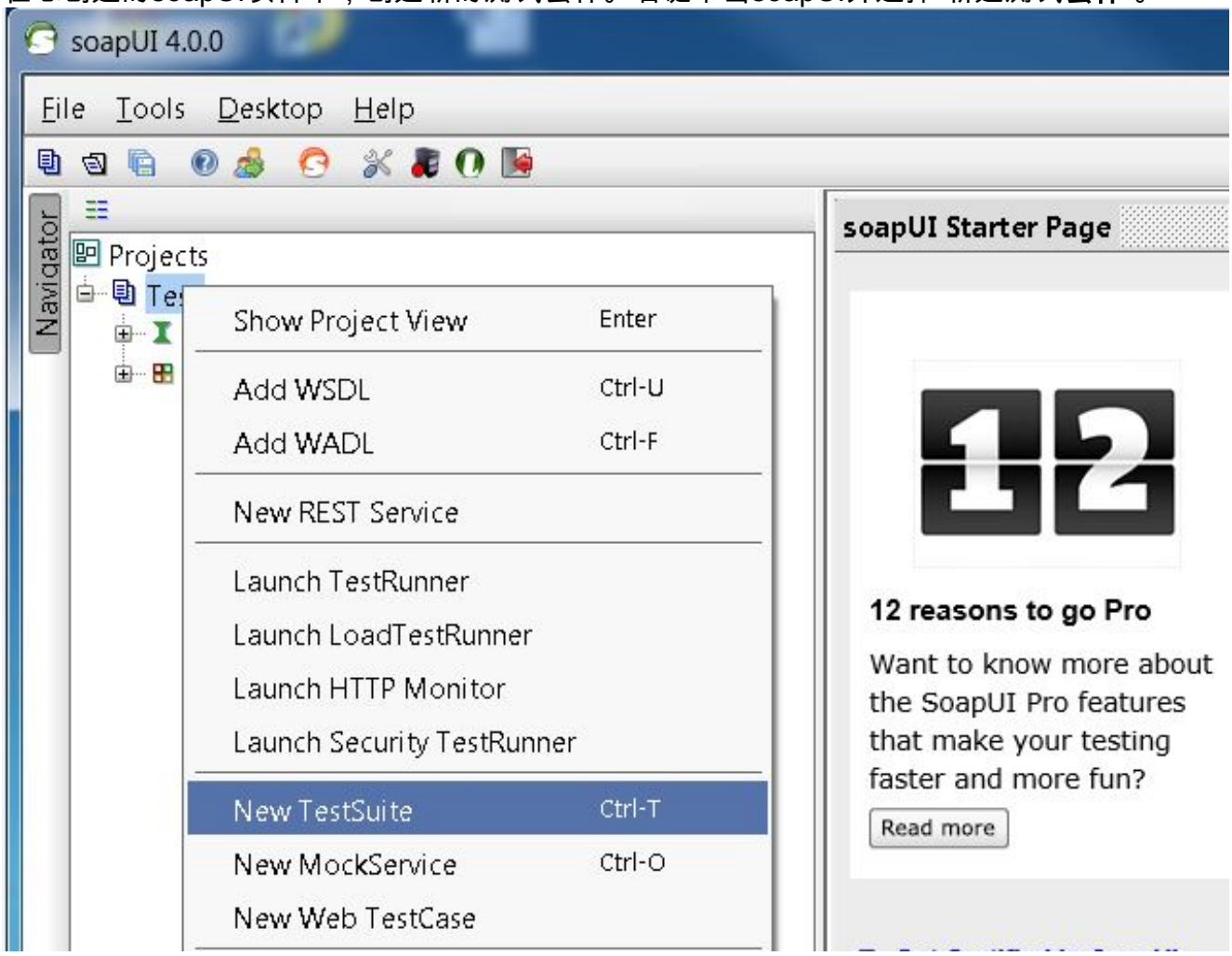
创建soapUI测试用例

此过程说明如何创建可在API发送到QPS时自动执行的测试套件。

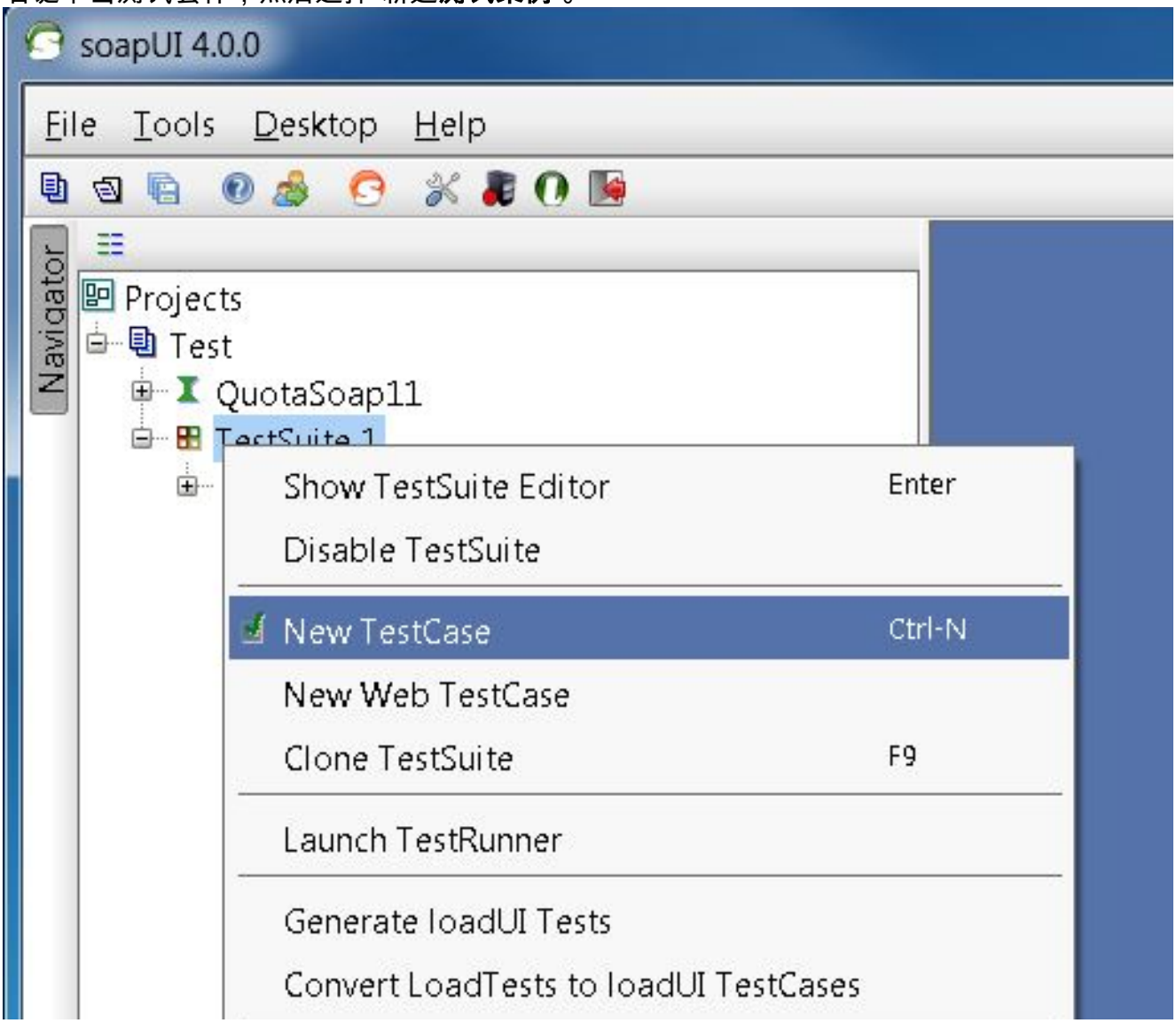
在本示例过程中，测试套件会循环访问用户ID列表，然后使用其发送到QPS的querySubscriberRequest中的这些用户ID。用户ID列表位于名为subid.txt的文本文件中的一行中。

要创建测试套件，请完成以下步骤：

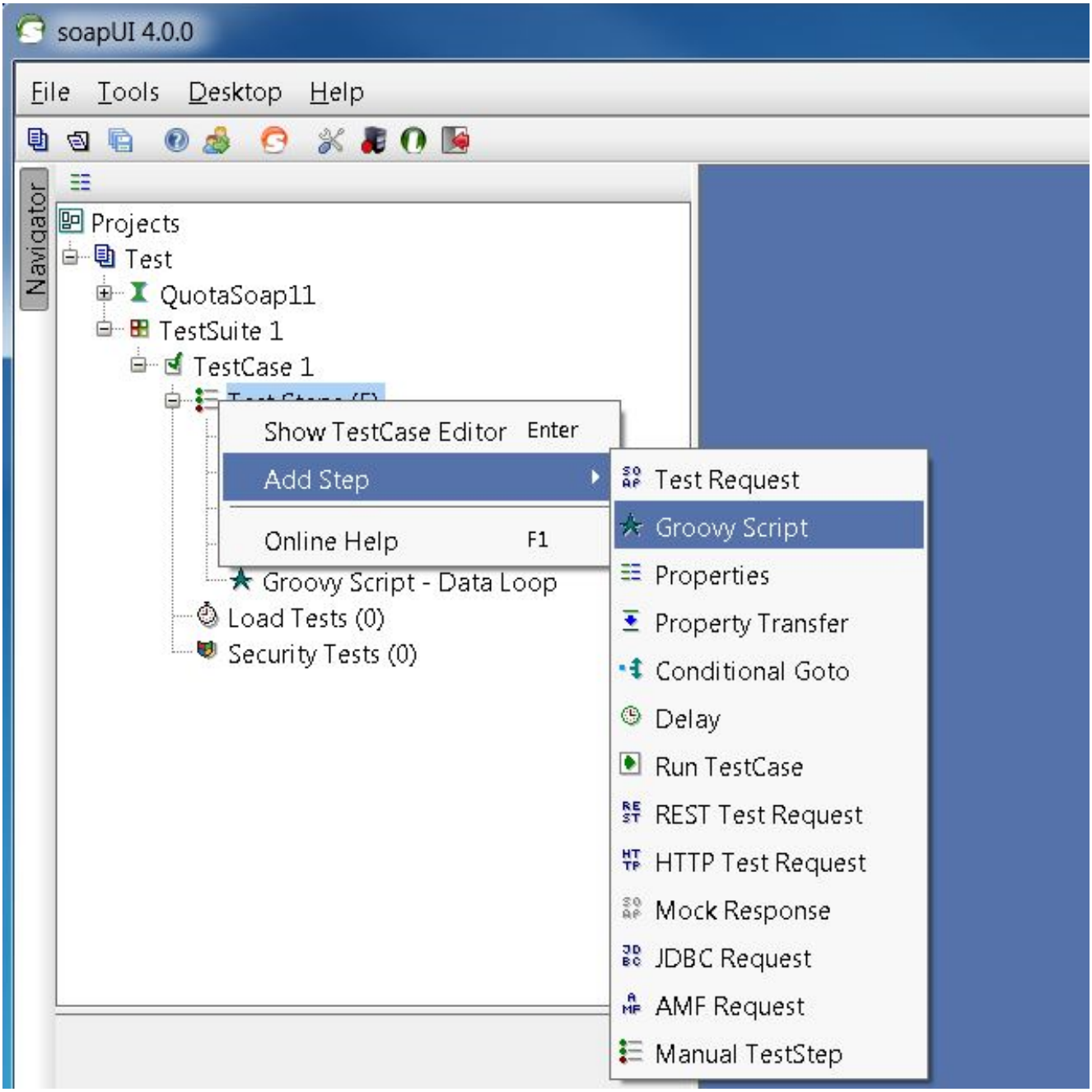
1. 在您创建的soapUI项目中，创建新的测试套件。右键单击soapUI并选择“新建测试套件”。



2. 右键单击测试套件，然后选择“新建**测试案例**”。



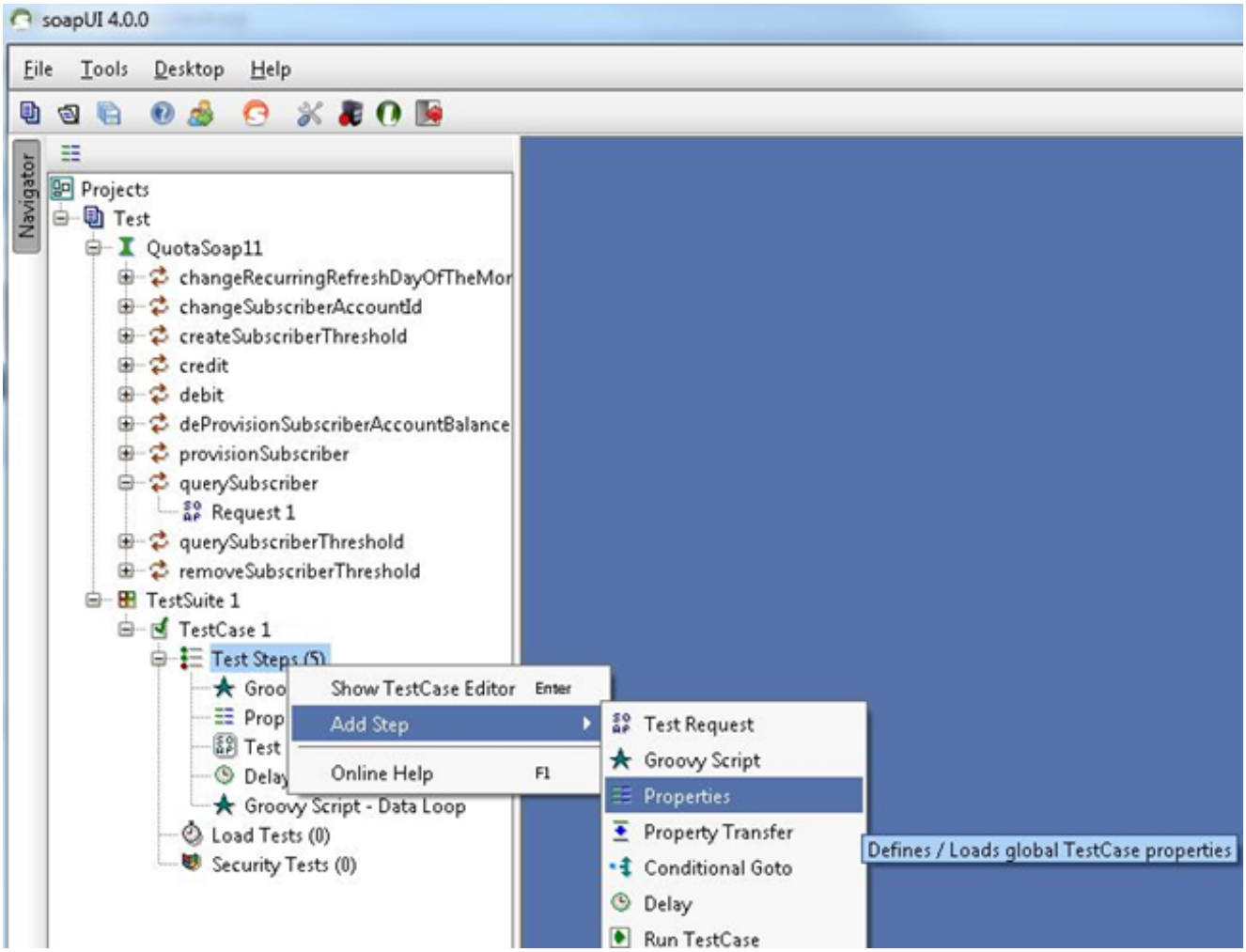
3. 右键单击“测试案例”，然后选择“**添加步骤**”> **Groovy脚本**以添加Groovy脚本测试步骤。将其命名为**数据源**
:



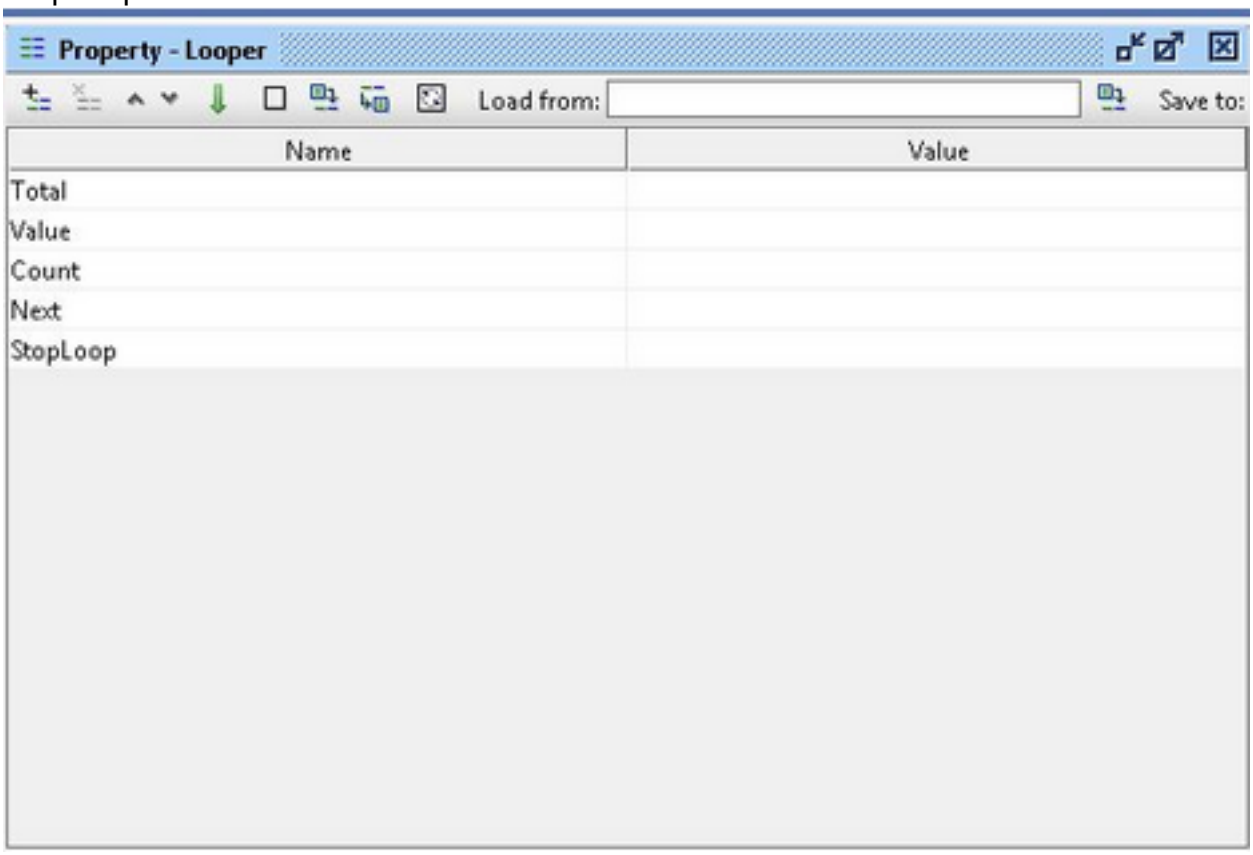
4. 在数据源文件中粘贴此代码。此代码读取文件**C:/subid.txt**，其中每行都包含用户ID:

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

5. 右键单击测试步骤并选择**添加步骤 > 属性**，以添加属性测试步骤并将其命名为属性 — 环回程序。

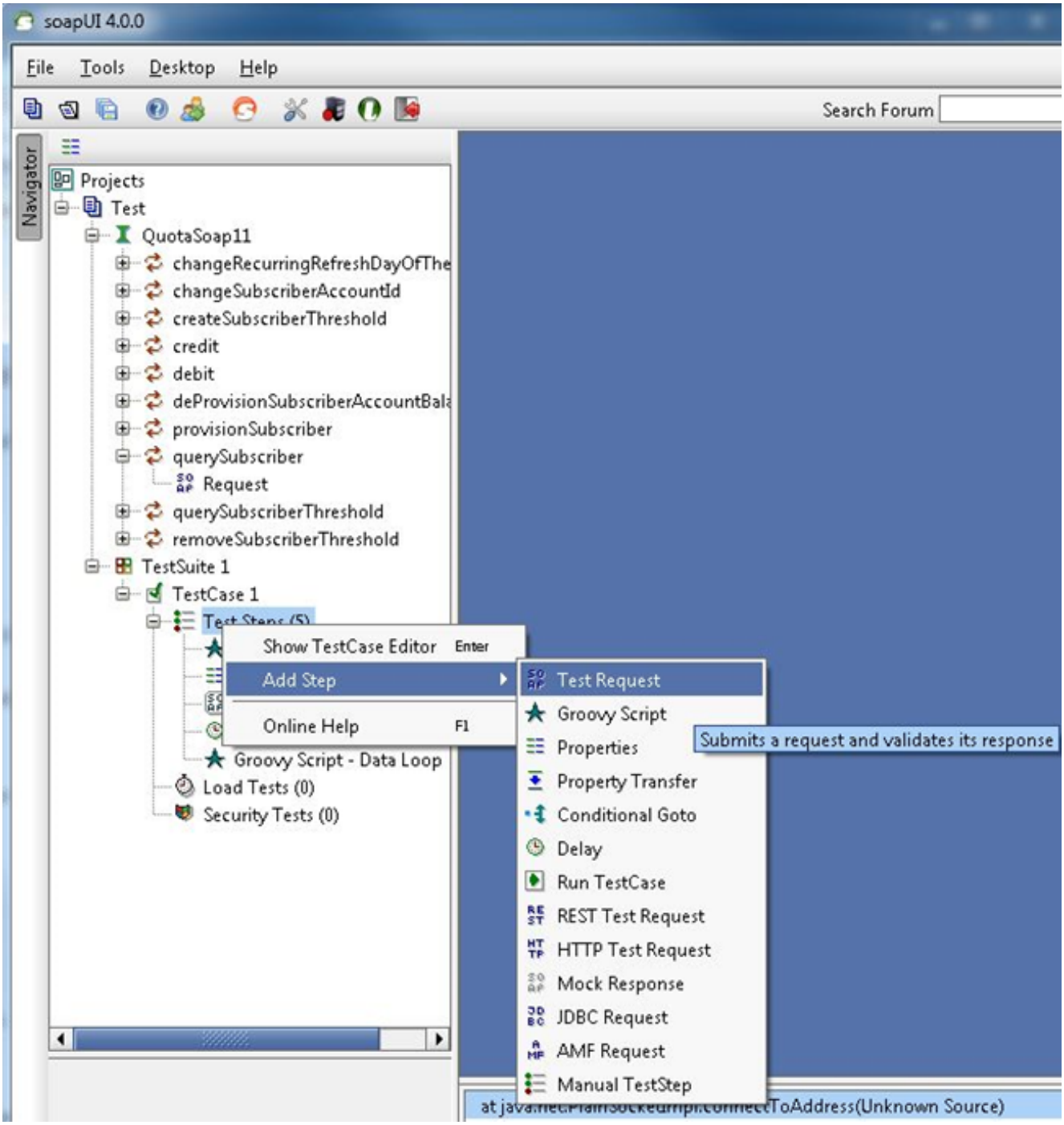


6. 添加Looper测试步骤的以下用户定义属性： 总数值 — 在本例中，保留从文件用户ID读取的用户ID计数下一步
   StopLoop

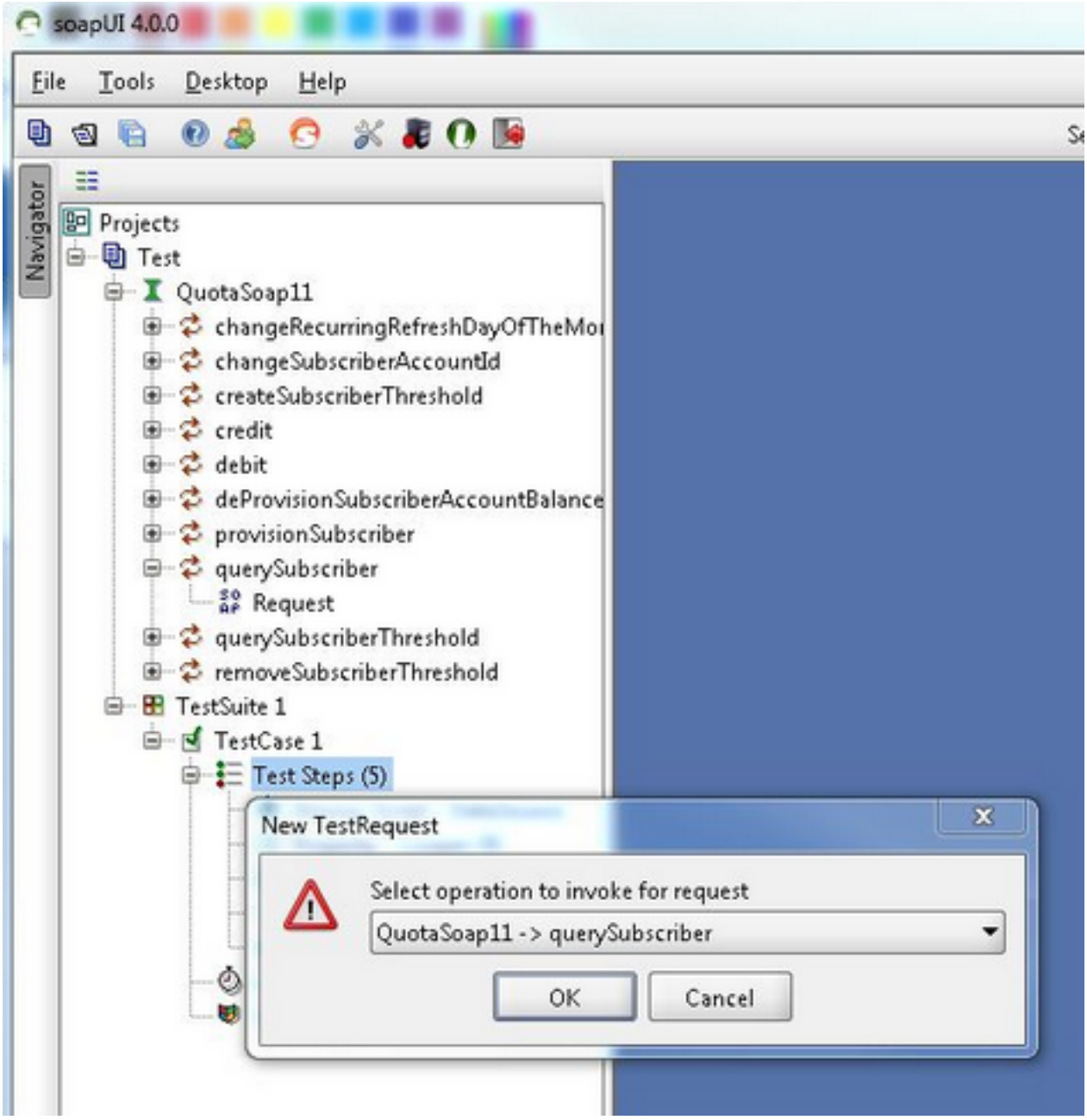


7. 右键单击测试步骤并选择**添加步骤 > 测试请求**以添加测试请求测试步骤并选择要调用的请求

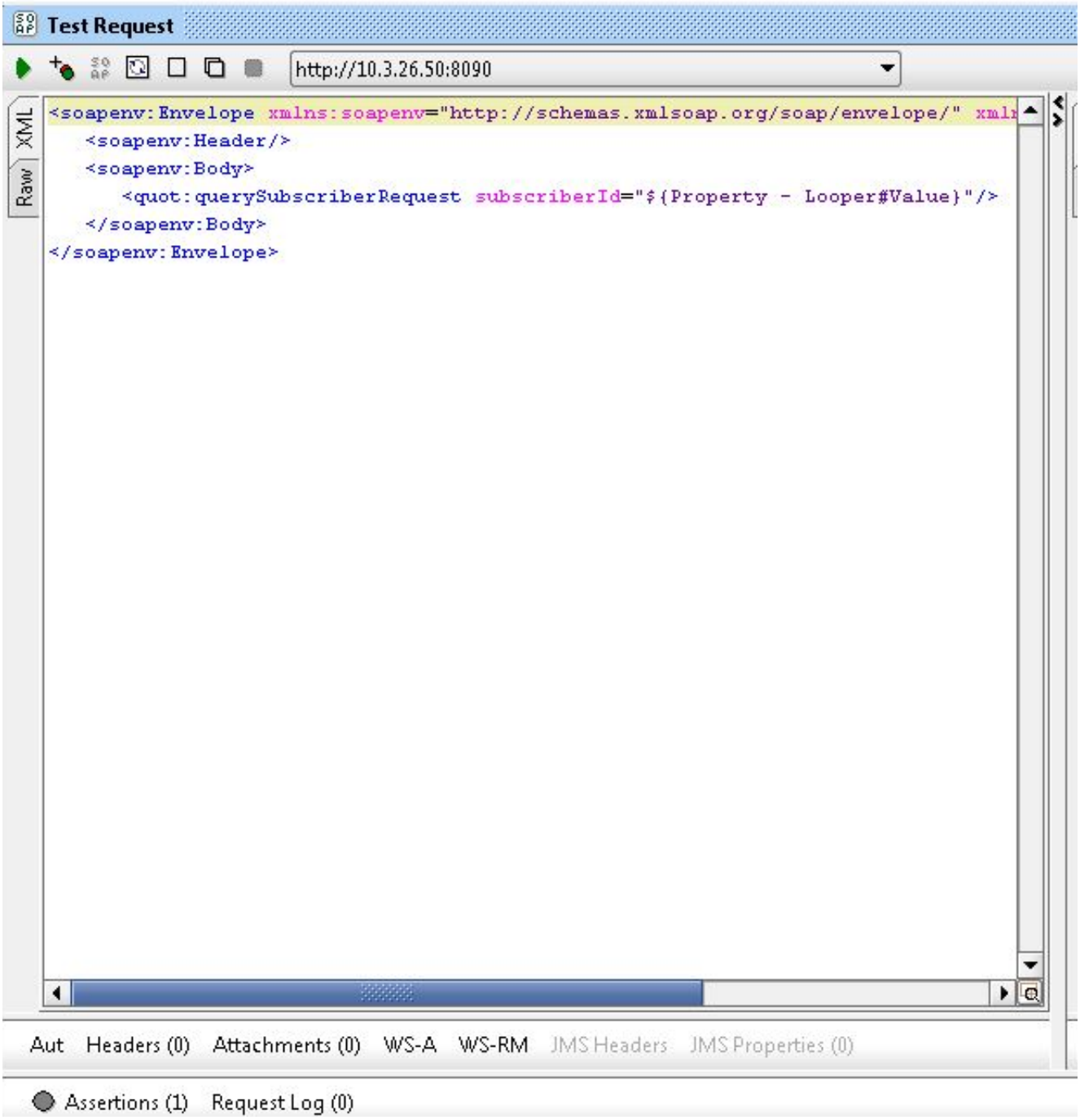
:



在本例中，使用querySubscriberRequest。

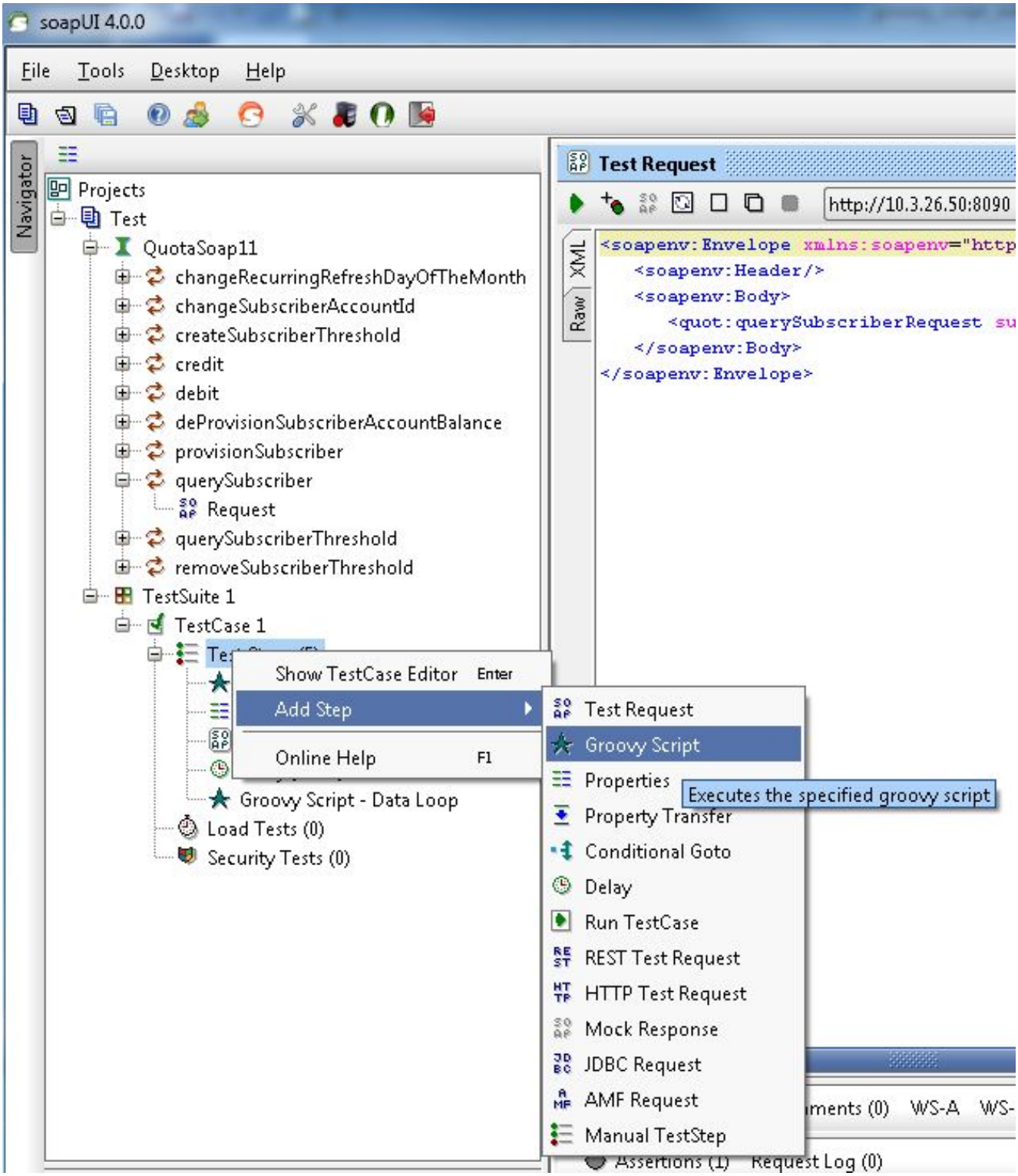


8. 在请求中，扩展代码将替换您查询的字段值。在本例中?的**IP地址？**在querySubscriberRequest中，将替换为扩展**代码${Property - Looper#Value}**(soap_test_req_expansion_code):



Property - Looper是先前创建的属性TestStep的名称，**Value**保存从用户ID文件读取的当前用户ID。

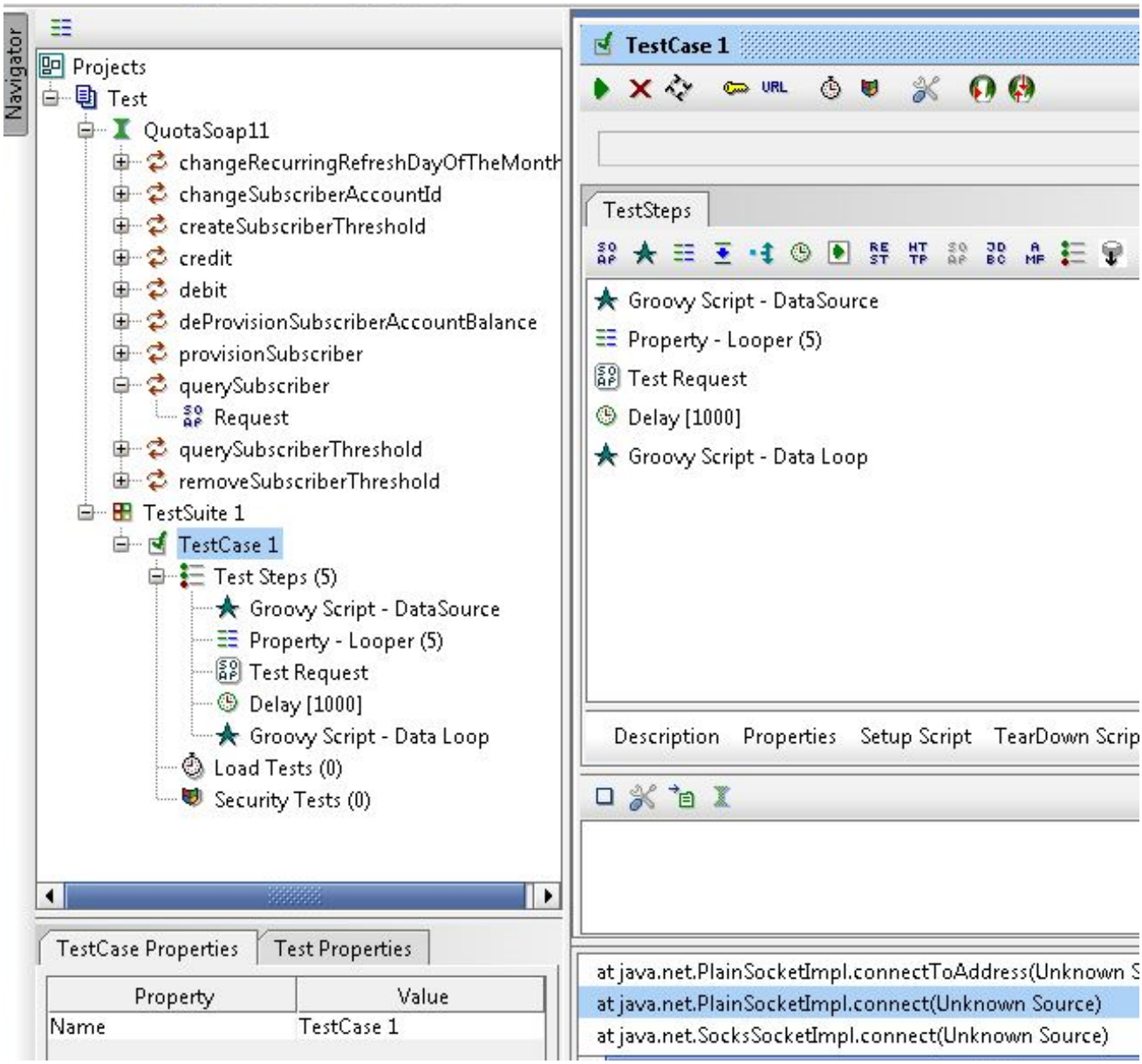
9. 右键单击测试步骤，然后选择**添加步骤> Groovy脚本**并将其命**名为Data Loop**:



10. 将此代码粘贴到Groovy脚本数据循环中：

```
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. 在本示例过程中，每个环路之间的延迟为1000毫秒。此步骤是可选的。由于延迟，现在有五个测试步骤
    ：



12. 单击绿色的**Run**按钮，以在TestCase窗口中运行五个测试步骤。