Automatiser les API avec Groovy Script

Contenu

Introduction

Ce document décrit comment créer une demande d'interface de programmation d'applications (API) soapUI et comment créer un cas de test soapUI qui effectue des boucles sur les étapes de test qui automatisent les demandes d'API vers Quantum Policy Suite (QPS).

L'exemple de cas de test soapUI de cet article met en oeuvre une procédure de test qui lit un fichier d'ID d'abonné, puis crée et envoie une requêteSubscriberRequst à QPS.

Créer un projet soapUI

Avant de commencer cette procédure, installez l'application soapUI sur votre bureau. Vous pouvez télécharger l'exécutable d'installation de soapUI à partir de www.soapui.org.

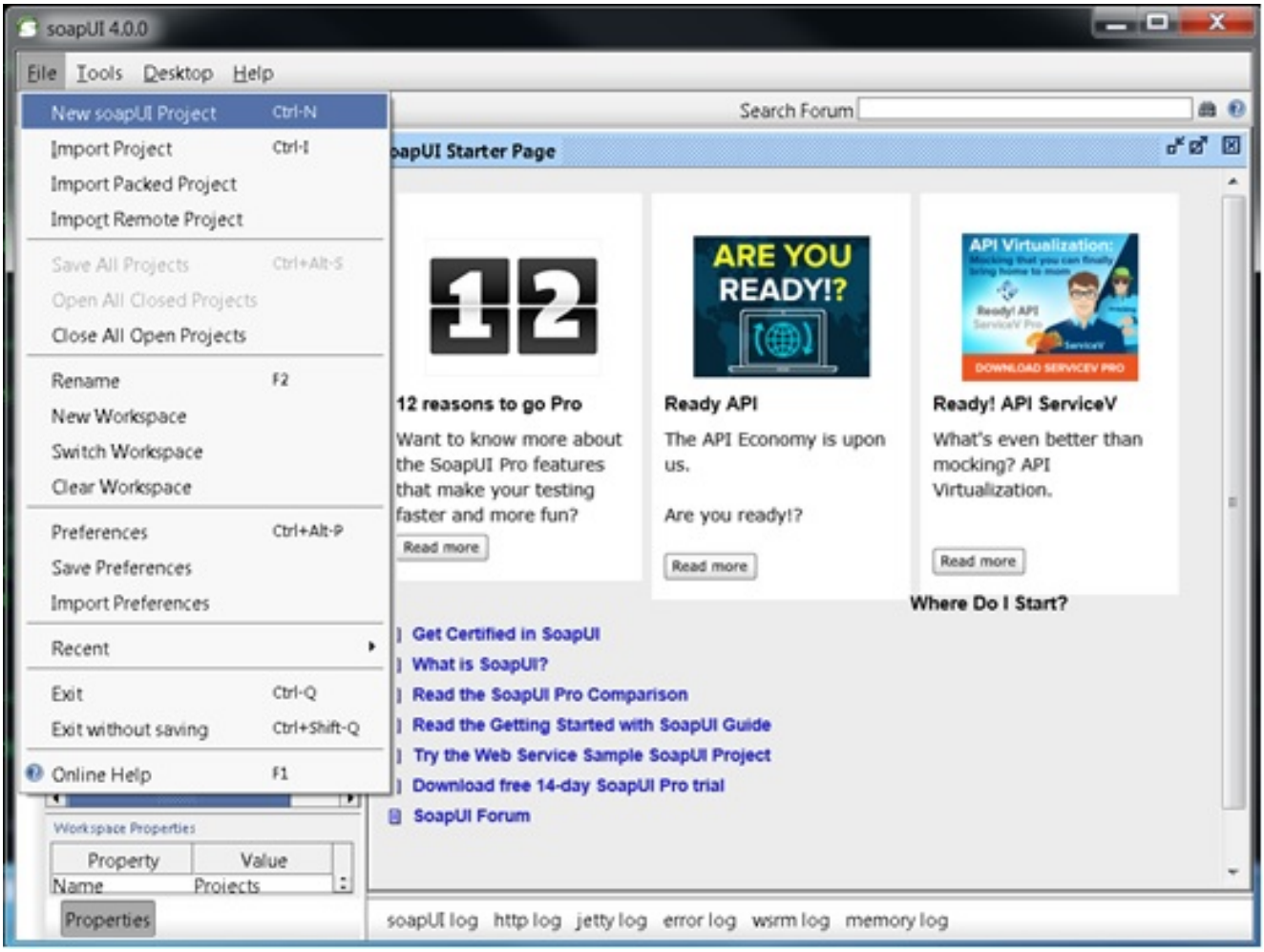
Avant de créer une demande d'API ou un cas de test, vous devez d'abord créer un projet soapUI. Vous avez besoin du fichier WSDL (Web Services Description Language) et du fichier XSD (XML Schema Description) pour créer le projet. Le WSDL spécifie les API prises en charge. Vous pouvez normalement obtenir le WSDL et le XSD du QPS lorsque vous exécutez ces commandes à partir de l'équilibrage de charge (LB) :

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

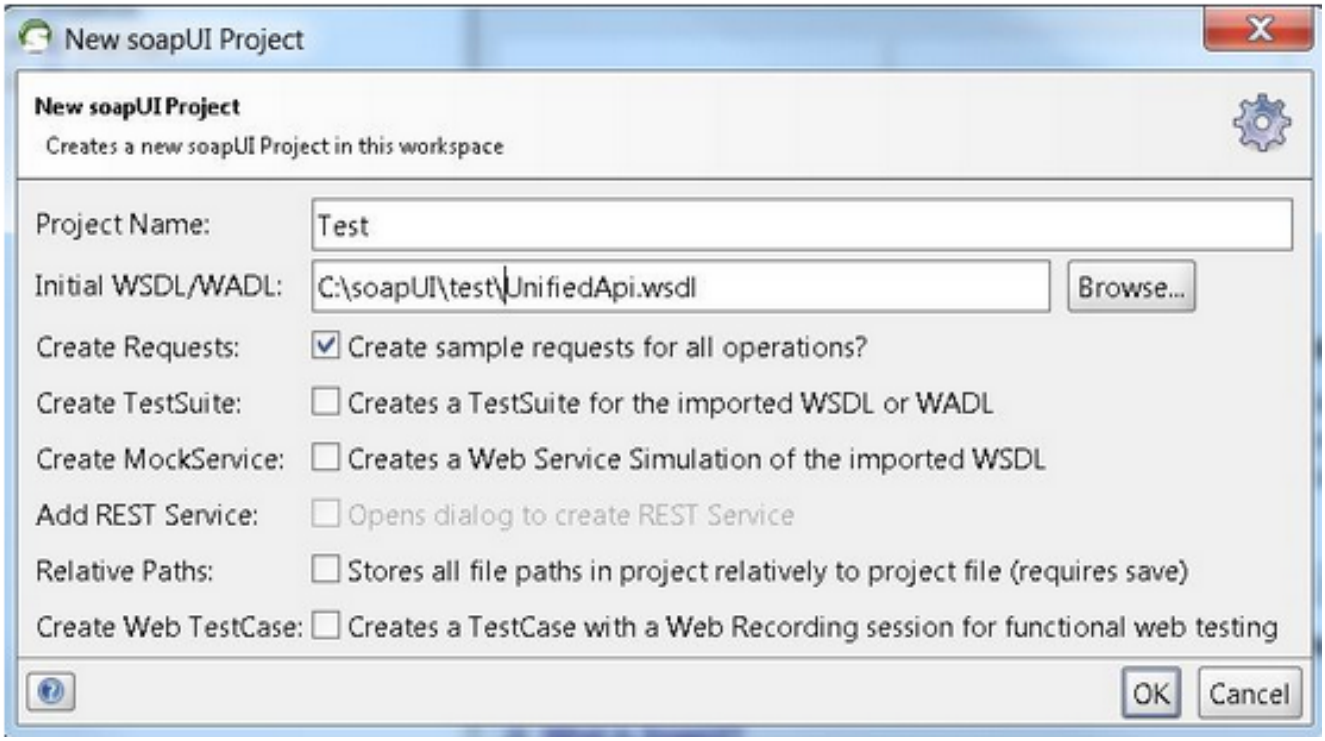
Stockez les fichiers WSDL et XSD dans le même répertoire sur le bureau où vous prévoyez d'exécuter l'application soapUI.

Complétez ces étapes afin de créer le projet soapUI :

1. Choisissez **Fichier > Nouveau projet soapUI** dans la fenêtre soapUI :

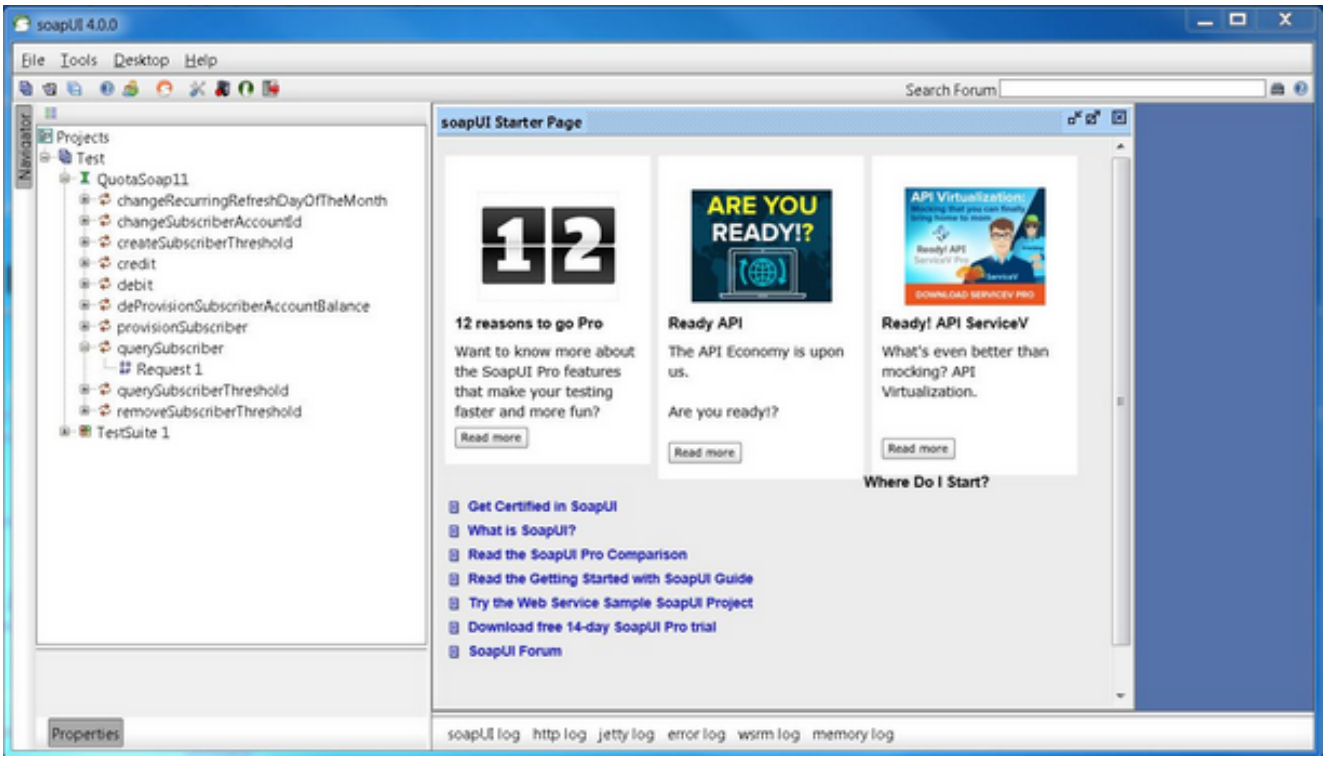


2. Dans la fenêtre Nouveau projet soapUI, saisissez un nom pour le projet dans le champ Nom du projet et saisissez l'emplacement où le fichier WSDL est stocké dans le champ Initial WSDL/WADL. Cliquez sur **OK lorsque vous avez terminé.**

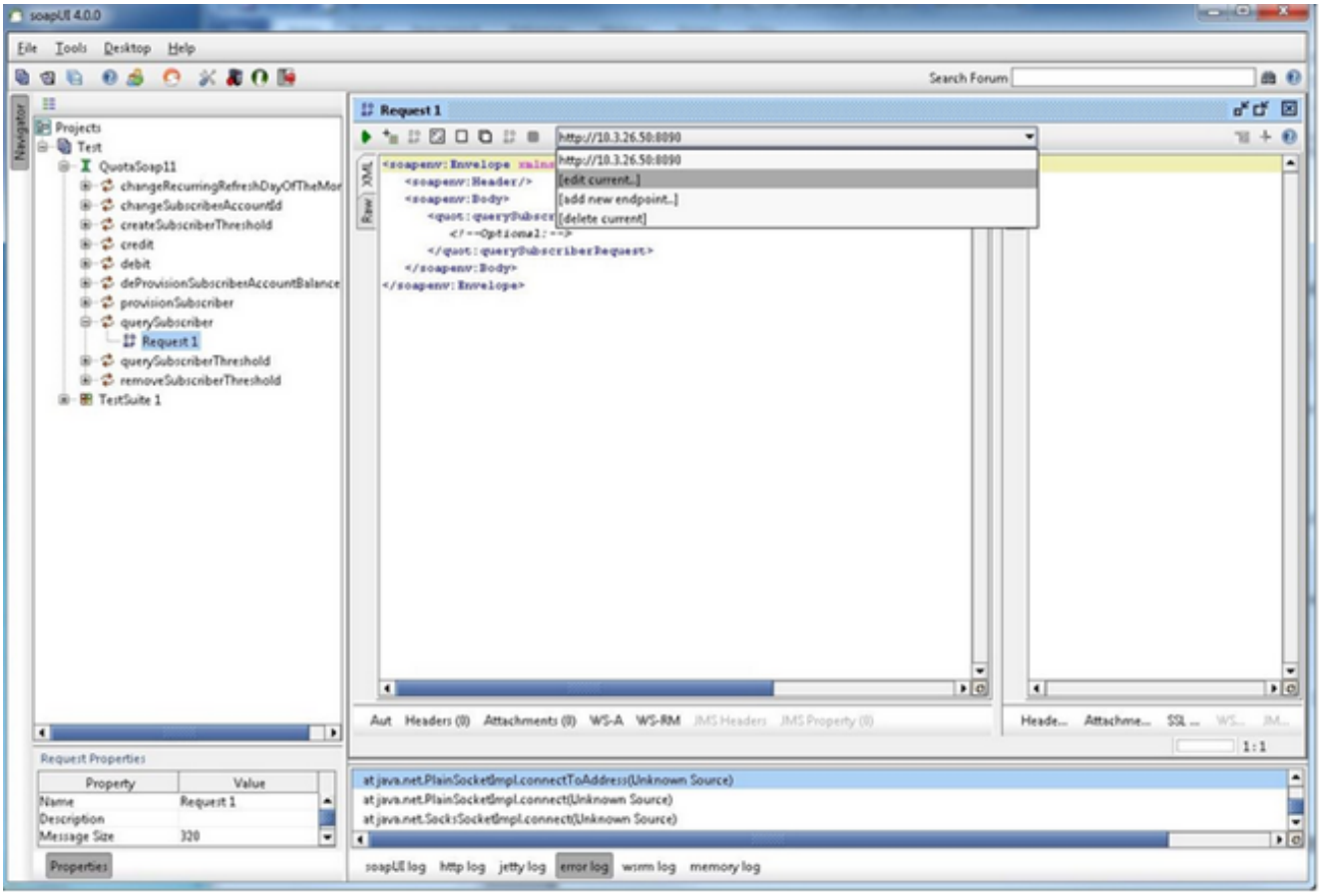


Créer une demande d'API soapUI

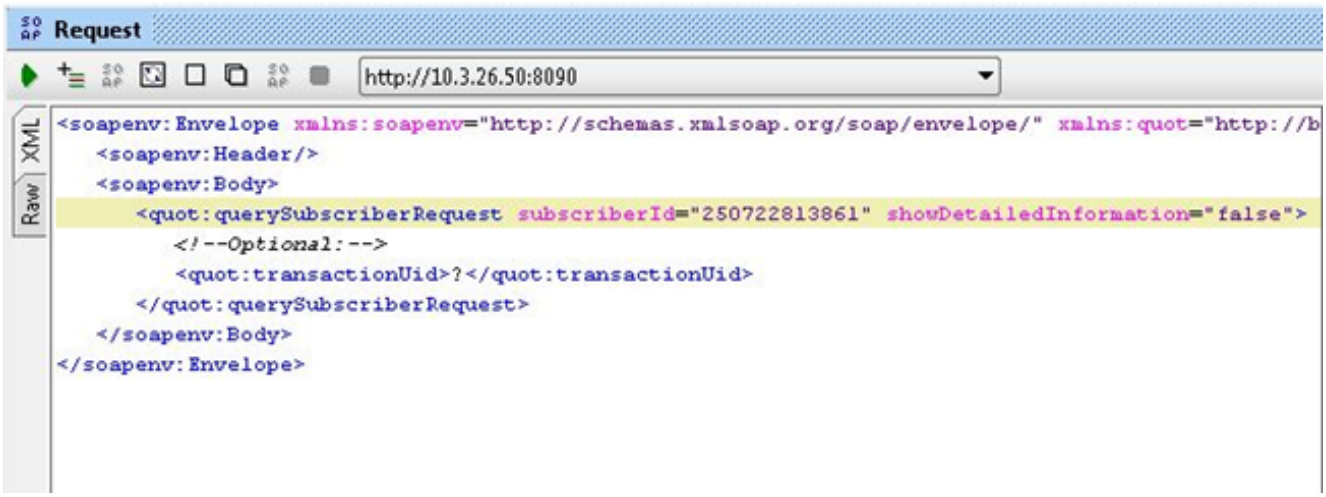
Complétez ces étapes afin de créer une demande d'API soapUI :

1. Développez le projet soapUI que vous avez créé afin de voir les API. Vous pouvez également développer une des API afin de voir la requête.Dans cet exemple, **querySubscriberRequest** est développé
   :
   

2. Ouvrez la requête afin de voir la fenêtre de requête avec le code XML qui forme la requête. Dans la fenêtre Requête, modifiez l'adresse IP http:// en adresse IP et port. Il s'agit généralement de l'adresse IP et du port lbvip01 où vous voulez envoyer la demande, comme le montre cet exemple
   :



3. Modifiez les champs du fichier XML avec les données que vous souhaitez envoyer dans votre demande. Dans cet exemple, la demande est querySubscriberRequest. Modifiez l'ID d'abonné de l'abonné que vous voulez interroger et définissez showDetailsInformation sur **false**
:



4. Cliquez sur le bouton vert **Exécuter** en haut de la fenêtre Demande afin d'exécuter la requête.

Créer un cas de test soapUI

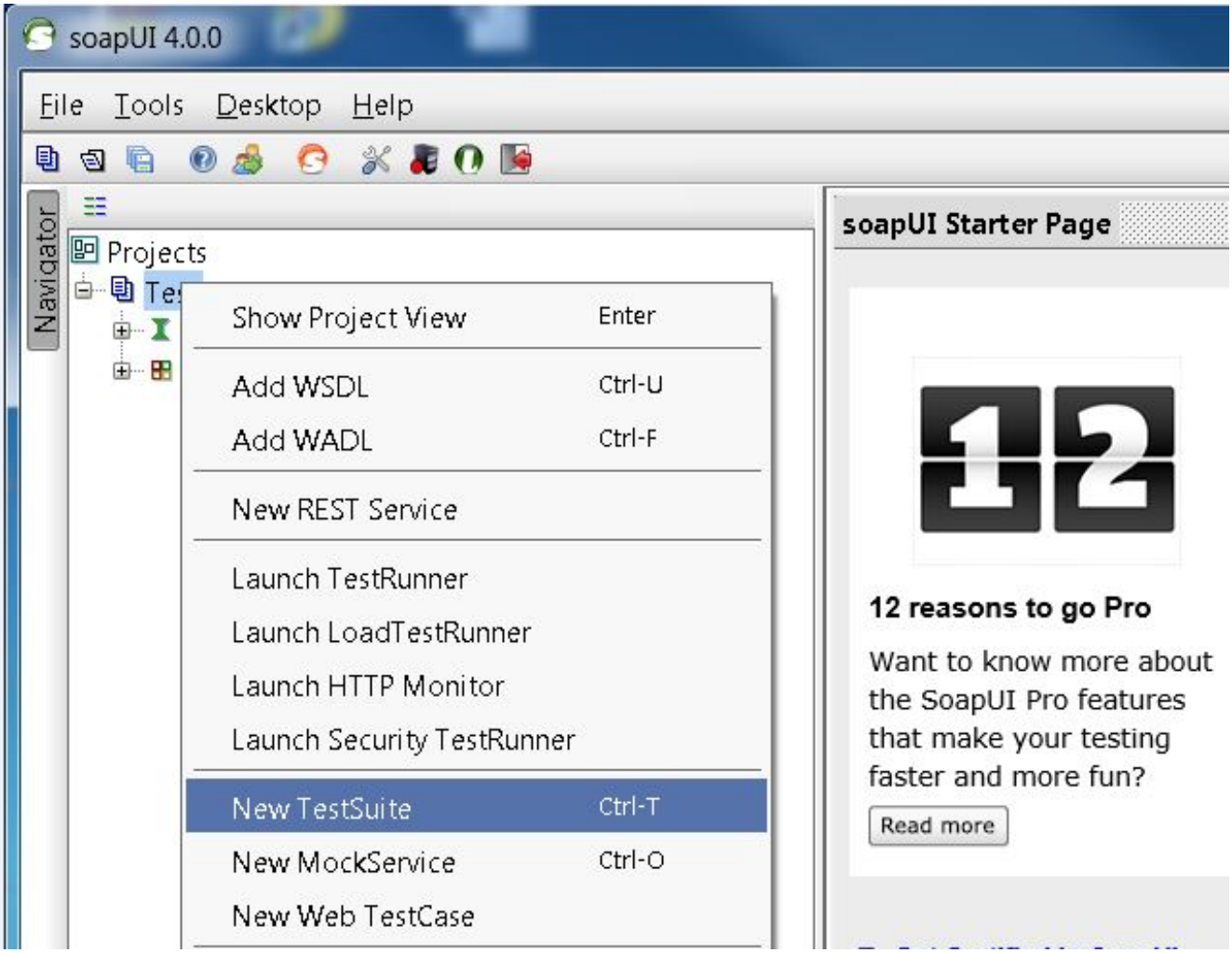
Cette procédure explique comment créer une suite de tests qui peut automatiser lorsque des API sont envoyées au QPS.

Dans cet exemple de procédure, la suite de tests effectue une boucle sur une liste d'ID d'abonné, puis utilise ces ID d'abonné dans la requêteSubscriberRequest qu'elle envoie à QPS. La liste des

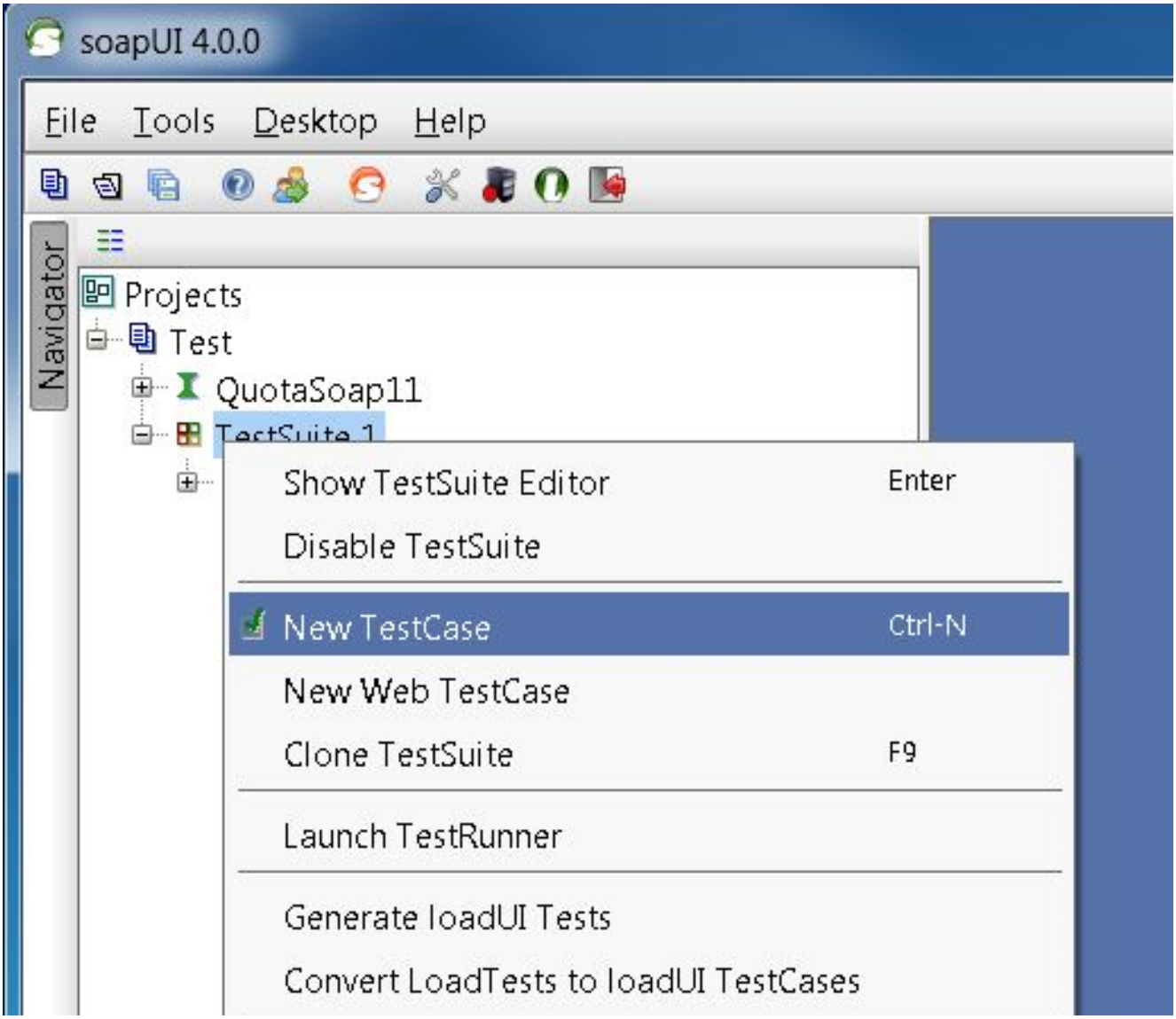
ID d'abonné se trouve sur une seule ligne d'un fichier texte appelé **subid.txt**.

Complétez ces étapes afin de créer la suite de tests :

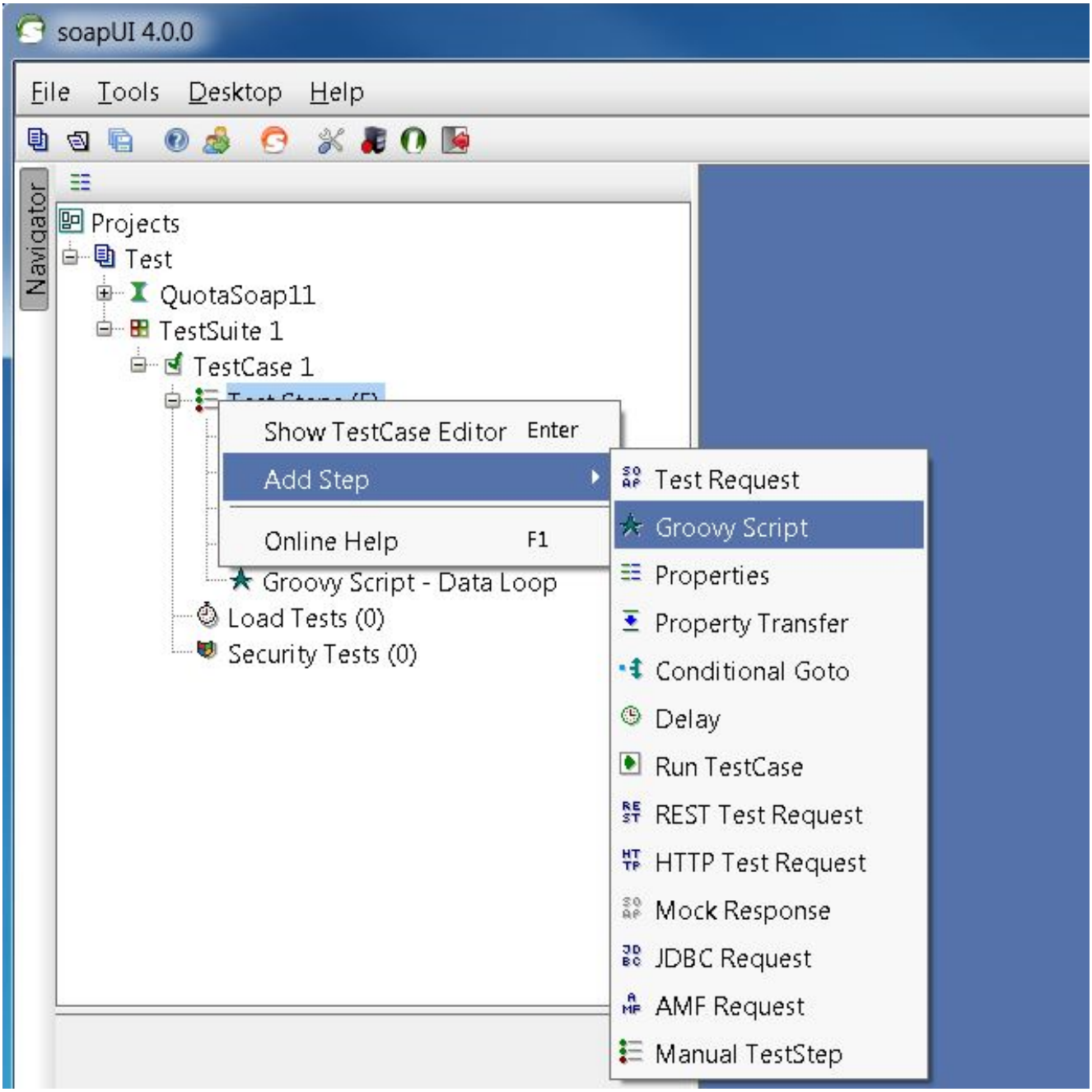
1. Dans le projet soapUI que vous avez créé, créez une nouvelle suite de tests. Cliquez avec le bouton droit sur le soapUI et choisissez **New TestSuite**.



2. Cliquez avec le bouton droit de la souris sur la suite de tests et sélectionnez **New TestCase**.



3. Cliquez avec le bouton droit sur le cas de test et choisissez **Ajouter une étape > Script Groovy** afin d'ajouter une étape de test Groovy Script. Nommer la **source de données** :

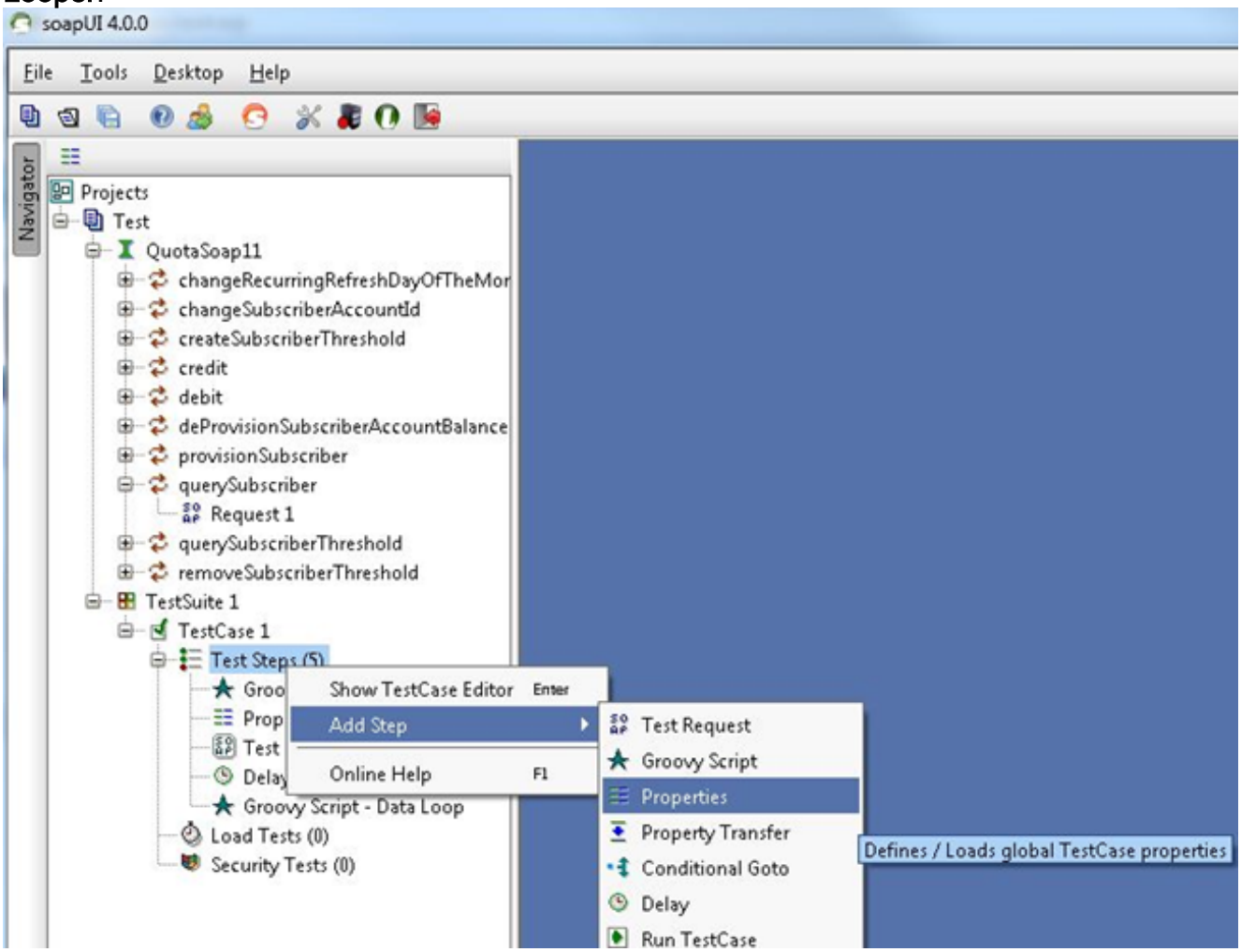


4. Dans le fichier source de données, collez ce code. Ce code lit le fichier **C : /subid.txt** qui contient un ID d'abonné sur chaque ligne :

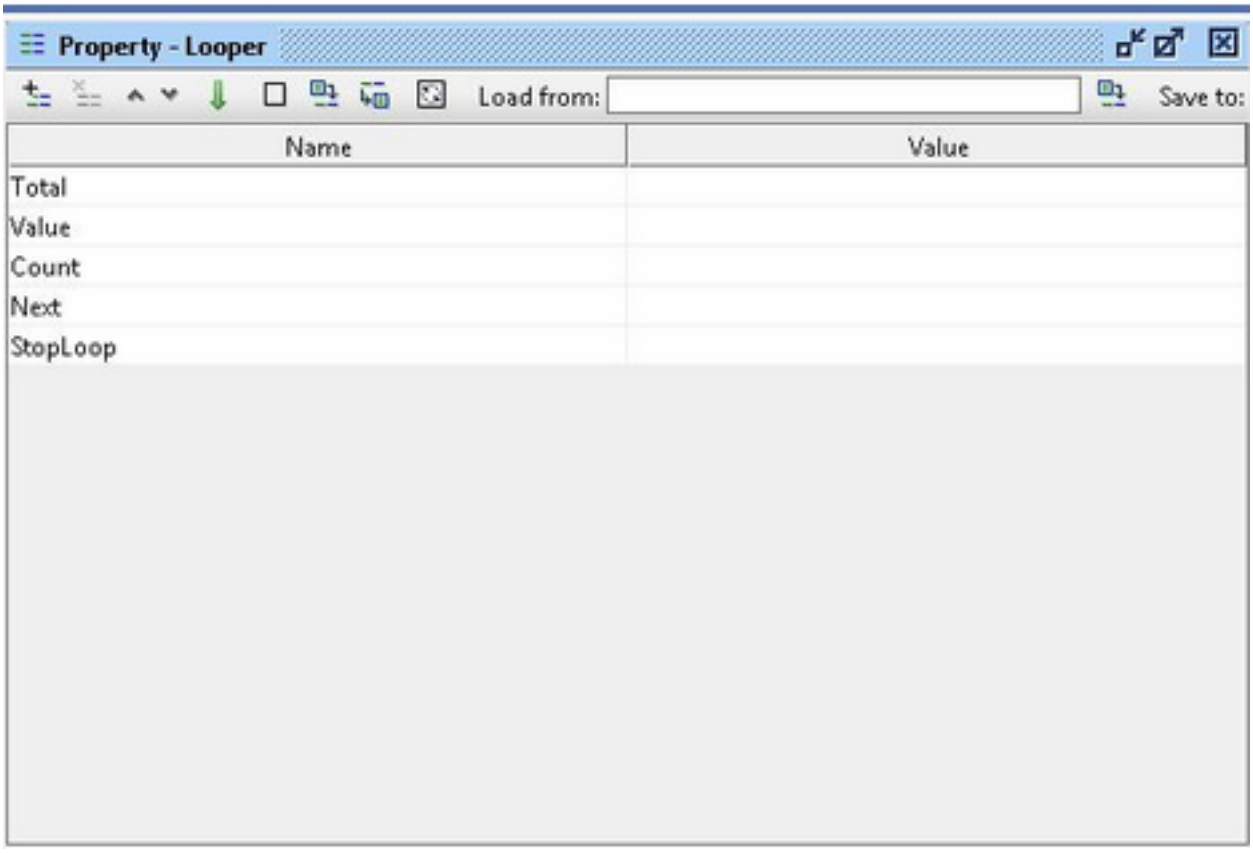
```
import com.eviware.soapui.support.XmlHolder def myTestCase = context.testCase
def counter,next,previous,sizeFile tickerEnumFile = new File("C:/subid.txt") //subscriber
IDs sepearted by new line (CR). List lines = tickerEnumFile.readLines() size =
lines.size.toInteger() propTestStep = myTestCase.getTestStepByName("Property - 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. Cliquez avec le bouton droit de la souris sur l'étape de test et choisissez **Ajouter une étape** > **Propriétés** afin d'ajouter une étape de test de propriété et de lui donner un nom **Propriété -**

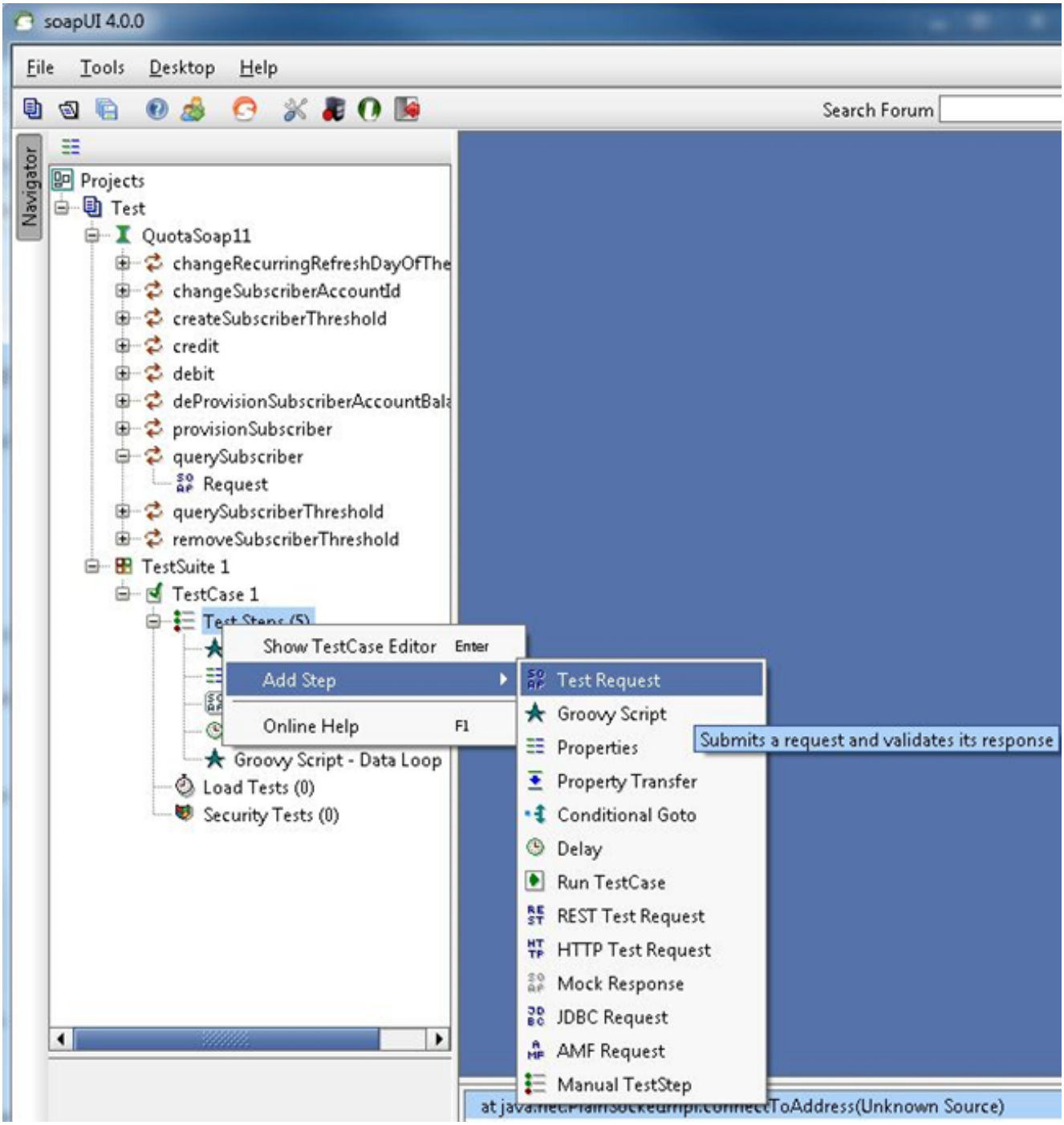
Looper.



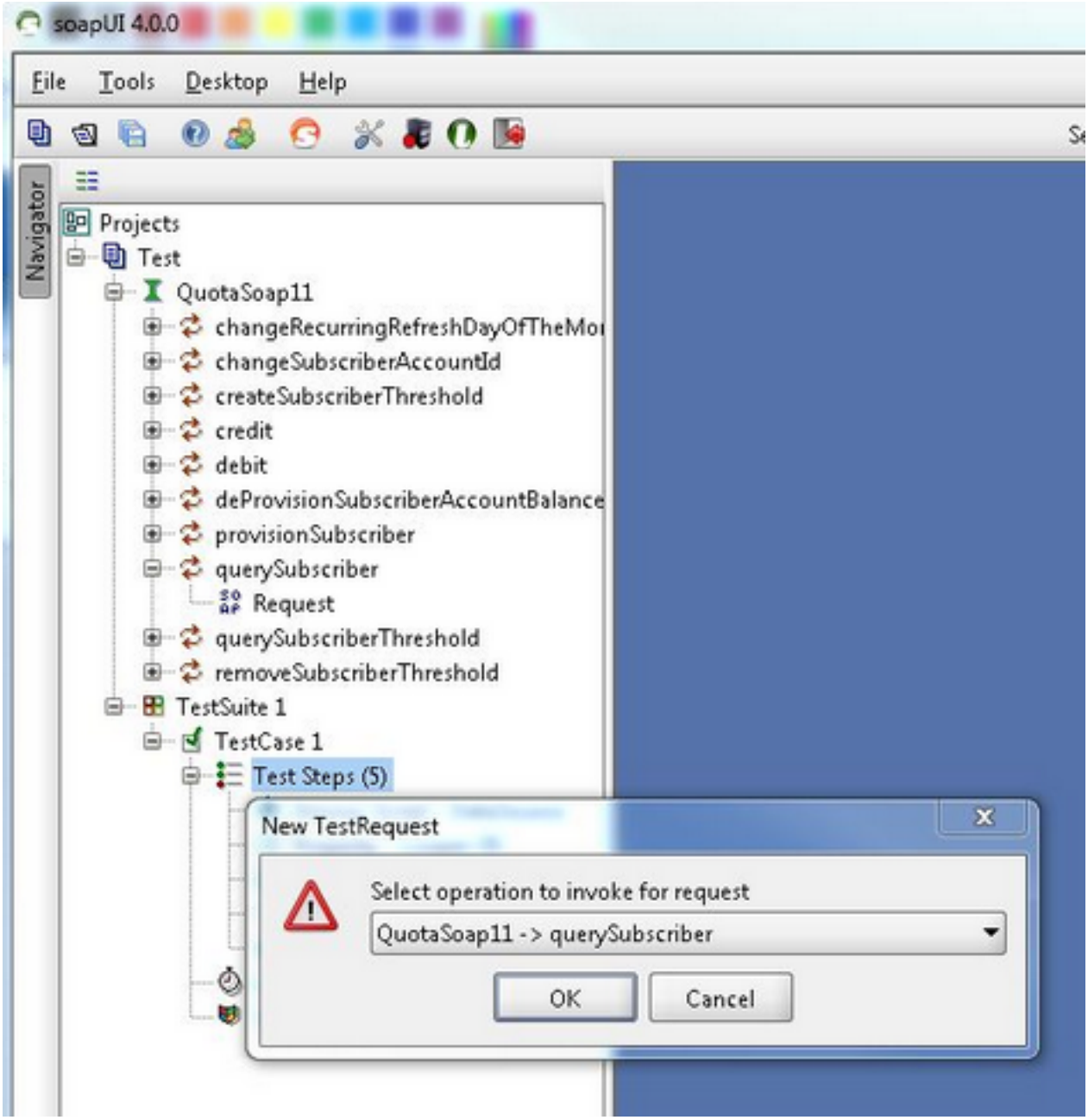
6. Ajoutez les propriétés définies par l'utilisateur de l'étape de test Looper : TotalValeur - Dans notre exemple, cette propriété contient l'ID d'abonné lu à partir des ID d'abonné du fichier.NombreSuivantArrêtBoucle



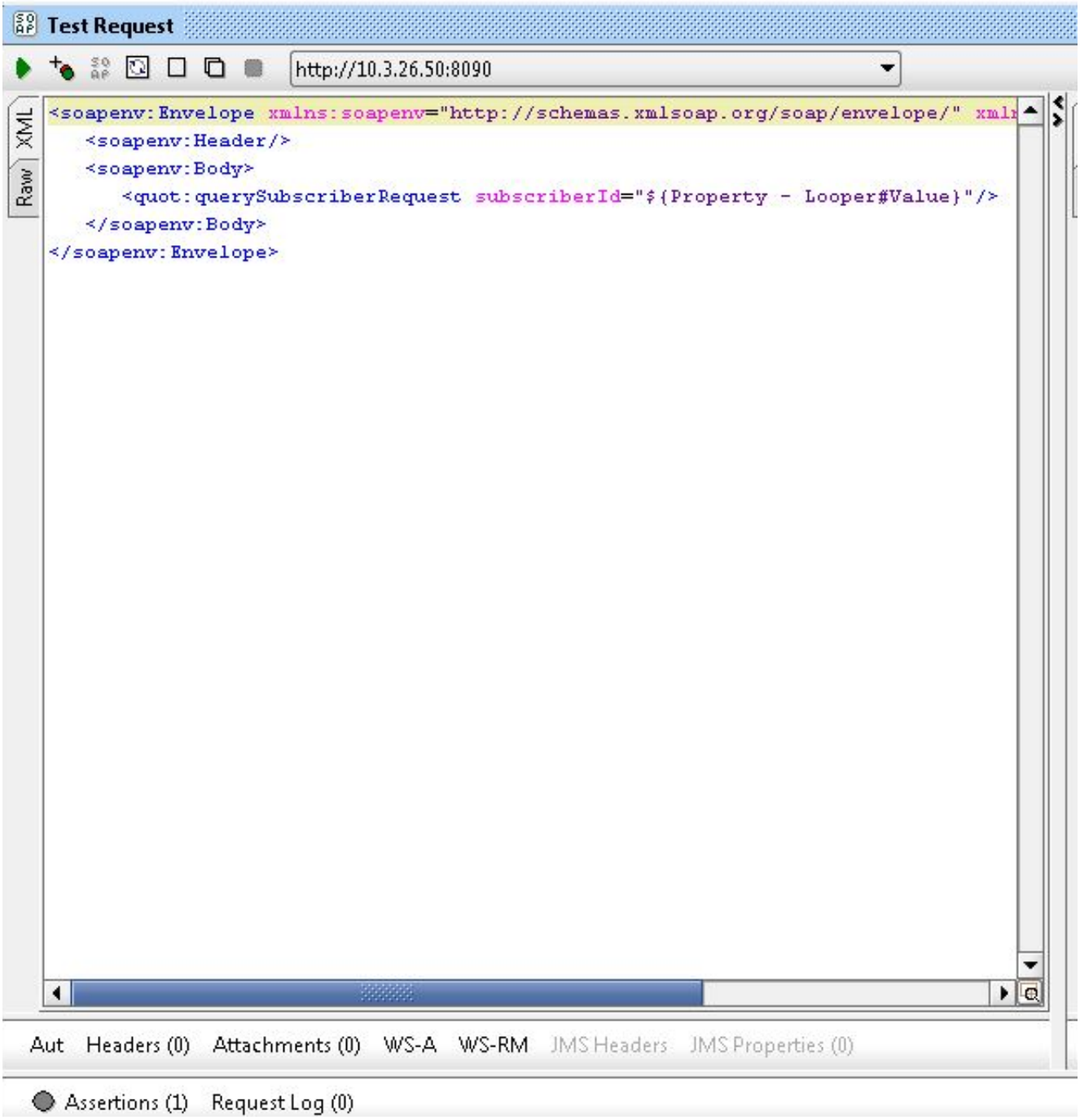
7. Cliquez avec le bouton droit sur l'étape de test et choisissez **Add Step** > **TestRequest** afin d'ajouter une étape de test de demande de test et choisissez la demande que vous voulez appeler
:



Dans cet exemple, querySubscriberRequest est utilisé.

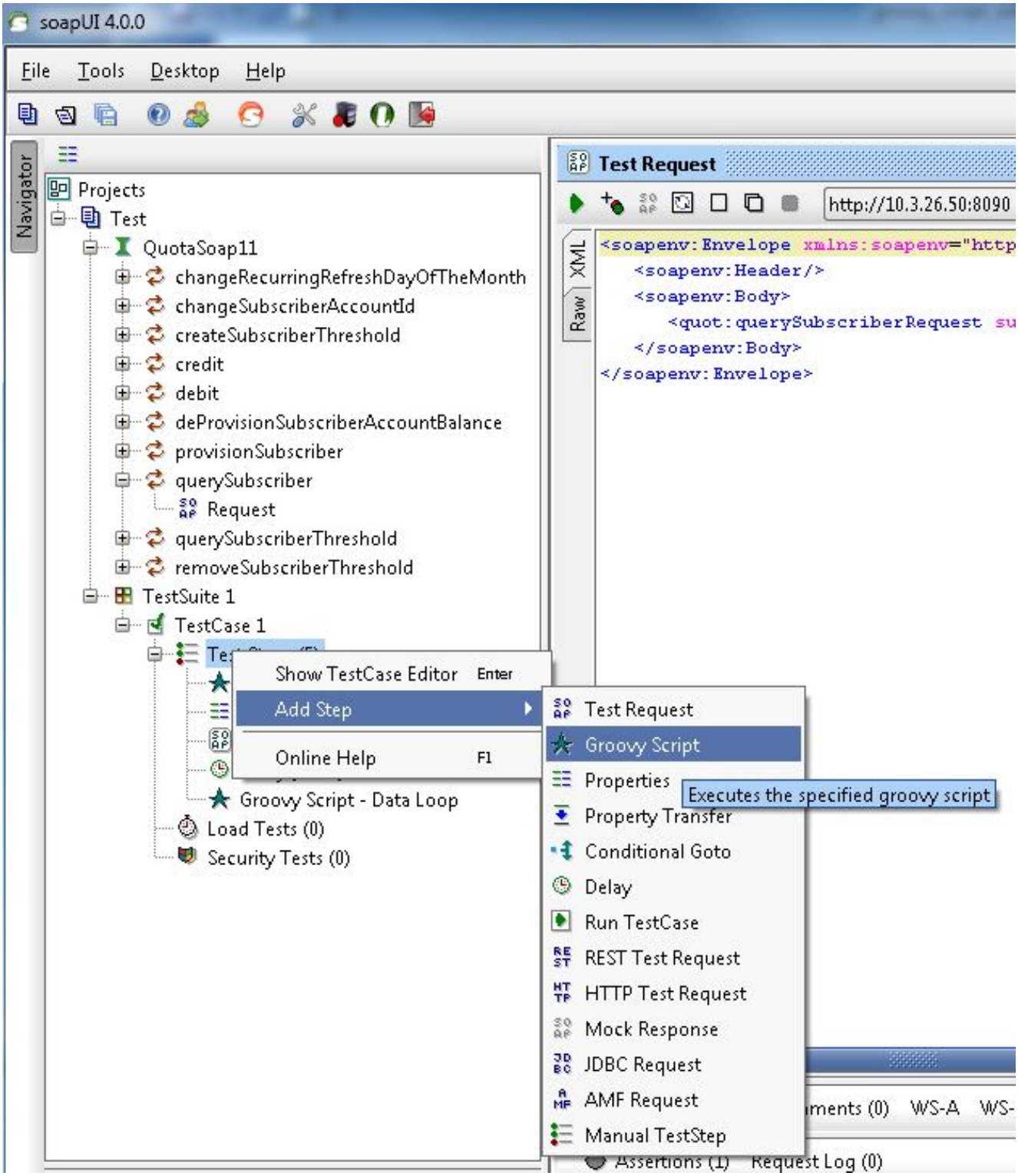


8. Dans la demande, le code d'extension remplace les valeurs de champ de ce que vous recherchez. Dans cet exemple, le **?** de **SubscriberId= ?** dans querySubscriberRequest est remplacé par le code d'extension **${Property - Looper#Value}** (soap_test_req_expansion_code)
   :



Propriété - Looper est le nom de Property TestStep précédemment créé et **Value** contient l'ID d'abonné actif lu à partir du fichier d'ID d'abonné.

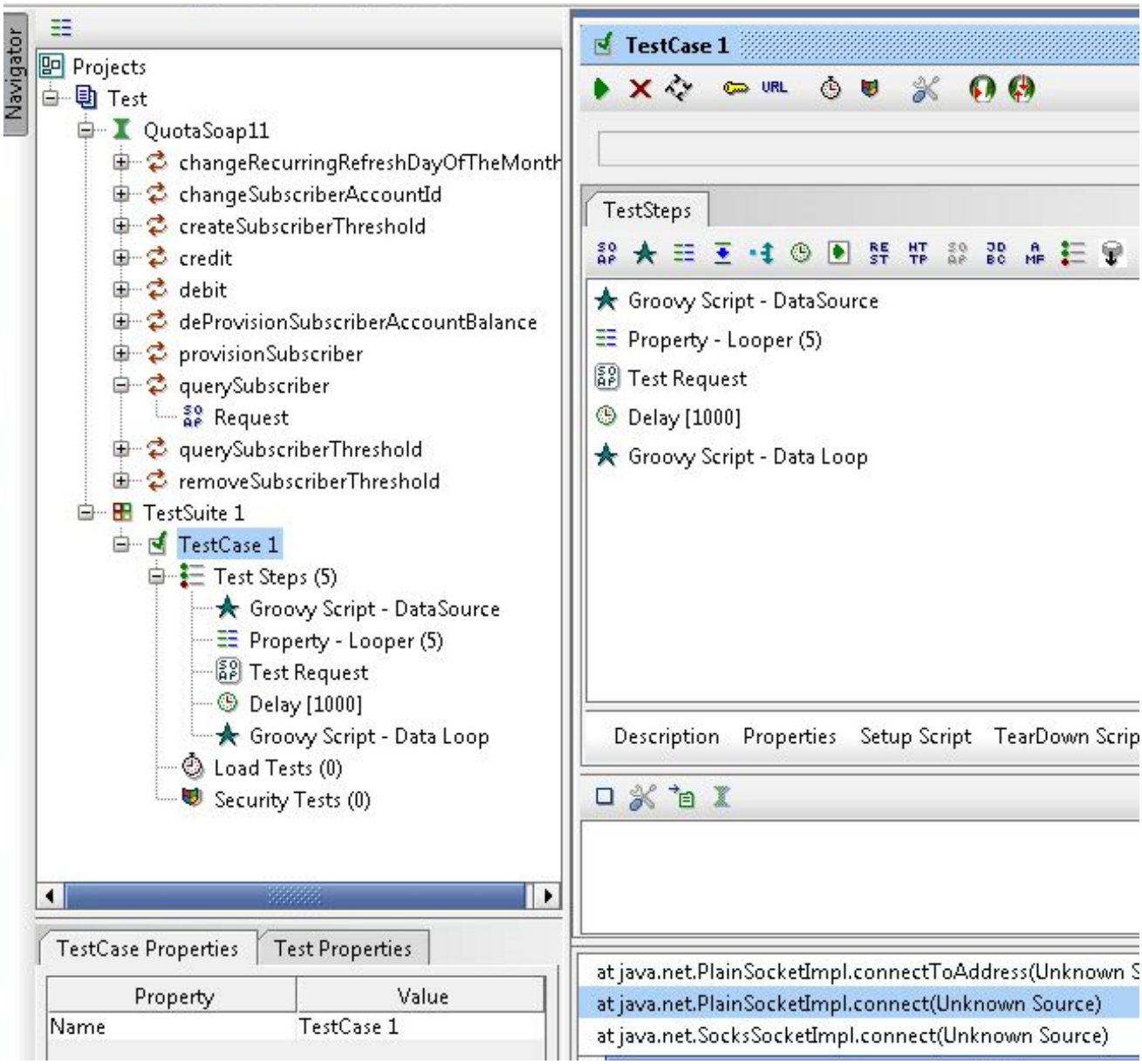
9. Cliquez avec le bouton droit de la souris sur l'étape de test et choisissez **Add Step > Groovy Script** et nommez-la **Boucle de données**
:



10. Collez ce code dans la boucle de données Groovy Script :

```
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. Dans cet exemple de procédure, un délai de 1 000 ms entre chaque boucle est ajouté. This step is optional.Avec le retard, il y a maintenant cinq étapes de test
:



12. Cliquez sur le bouton **Exécuter** vert afin d'exécuter les cinq étapes de test dans la fenêtre TestCase.