

# Automatisez les API avec le script routinier

## Contenu

[Introduction](#)

[Créez un projet de soapUI](#)

[Créez une demande du soapUI API](#)

[Créez un cas de test de soapUI](#)

## Introduction

Ce document décrit comment créer des programmeurs d'une application de soapUI reliant la demande (API) et comment créer un cas de test de soapUI que les boucles au-dessus des étapes de test qui automatisent l'API demande à la suite de stratégie de Quantum (QPS).

Le cas de test de soapUI d'exemple dans les étapes de ce d'article test d'implementa qui lisent un fichier des id d'abonné et puis créent et envoient un querySubscriberRequst à QPS.

## Créez un projet de soapUI

Avant que vous commenciez cette procédure, installez l'application de soapUI sur votre appareil de bureau. Vous pouvez télécharger l'installation de soapUI exécutable de [www.soapui.org](http://www.soapui.org).

Avant que vous créiez une demande ou le cas de test API vous devez d'abord créer un projet de soapUI. Vous avez besoin du fichier du langage WSDL (WSDL) et du fichier de la description de schéma XML (XSD) afin de créer le projet. Le WSDL spécifie les API pris en charge. Vous pouvez normalement obtenir le WSDL et le XSD du QPS quand vous exécutez ces commandes de l'équilibreur de charge (livre) :

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

Enregistrez le WSDL et le XSD dans le même répertoire sur l'appareil de bureau où vous prévoyez d'exécuter l'application de soapUI.

Terminez-vous ces étapes afin de créer le projet de soapUI :

1. Choisissez le **fichier > nouveau projet de soapUI de la** fenêtre de soapUI :
2. Dans la nouvelle fenêtre de projet de soapUI écrivez un nom pour le projet dans le domaine de nom du projet et entrez l'emplacement où le fichier WSDL est enregistré dans le domaine de l'initiale WSDL/WADL. Cliquez sur **OK** quand vous avez terminé.

## Créez une demande du soapUI API

Terminez-vous ces étapes afin de créer une demande du soapUI API :

1. Développez le projet de soapUI que vous avez créé afin de voir les API. Vous pouvez également développer un des API afin de voir la demande. Dans cet exemple, **querySubscriberRequest** est développé :
2. Ouvrez la demande afin de voir la fenêtre de demande avec le XML qui forme la requête. Dans la fenêtre de demande éditez l'adresse IP de http:// à l'adresse IP et au port. C'est normalement l'adresse IP lbvip01 et le port où vous voulez envoyer le comme indiqué dans cet exemple de demande :
3. Modifiez les champs dans le XML avec les données que vous voulez introduire votre demande. Dans cet exemple, la demande est une querySubscriberRequest. Modifiez l'ID d'abonné pour l'abonné que vous voulez questionner et placer le showDetailedInformatin à **faux** :
4. Cliquez sur le bouton vert de **passage** en haut de la fenêtre de demande afin d'exécuter la requête.

## Créez un cas de test de soapUI

Cette procédure explique comment créer une suite de test qui peut automatiser quand des API sont envoyés au QPS.

Dans cet exemple de procédure la suite de test fait une boucle au-dessus d'une liste d'id d'abonné et puis utilise ces id d'abonné dans le querySubscriberRequest qu'elle envoie à QPS. La liste d'id d'abonné sont chacune sur une ligne simple dans un fichier texte appelé **subid.txt**.

Terminez-vous ces étapes afin de créer la suite de test :

1. Dans le projet de soapUI que vous avez créé, créez une nouvelle suite de test. Cliquez avec le bouton droit sur le soapUI et choisissez **nouveau TestSuite**.
2. Cliquez avec le bouton droit la suite de test et choisissez le **nouveau jeu d'essai**.
3. Cliquez avec le bouton droit le cas de test et choisissez **ajoutent l'étape > script routinier** afin d'ajouter une étape routinière de test de script. Nommez-le **point d'émission de données** :
4. Dans la pâte de fichier de point d'émission de données ce code. Ce code lit le **C de fichier** :

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

5. Cliquez avec le bouton droit sur l'étape de test et choisissez **ajoutent l'étape > le Properties**

afin d'ajouter une étape de test de propriété et la nommer **propriété - Looper**.

6. Ajoutez ces propriétés définies par l'utilisateur de l'étape de test de Looper : TotalValeur - Dans notre exemple cette attente l'ID d'abonné a indiqué des id d'abonné de fichierCompteEnsuiteStopLoop
7. Cliquez avec le bouton droit sur l'étape de test et choisissez **ajoutent l'étape > le TestRequest** afin d'ajouter une étape de test de demande de test et choisir 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 questionnez. Dans cet exemple ? du **SubscriberId= ?** dans le querySubscriberRequest est remplacé par le code **`{propriété d'extension - Looper#Value}`** (soap\_test\_req\_expansion\_code) : **Propriété - Le Looper** est le nom de la propriété TestStep précédemment créé et la **valeur** juge l'ID en cours d'abonné lu à partir du fichier des id d'abonné.
9. Cliquez avec le bouton droit sur l'étape de test et choisissez **ajoutent l'étape > script routinier** et la nomment **boucle de données** :
10. Collez ce code dans la boucle routinière de données de 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 retard du ms 1000 entre chaque boucle est ajouté. Cette étape est facultative. Avec le retard il y a maintenant cinq étapes de test :
12. Cliquez sur le bouton vert de **passage** afin d'exécuter les cinq étapes de test dans la fenêtre de jeu d'essai.