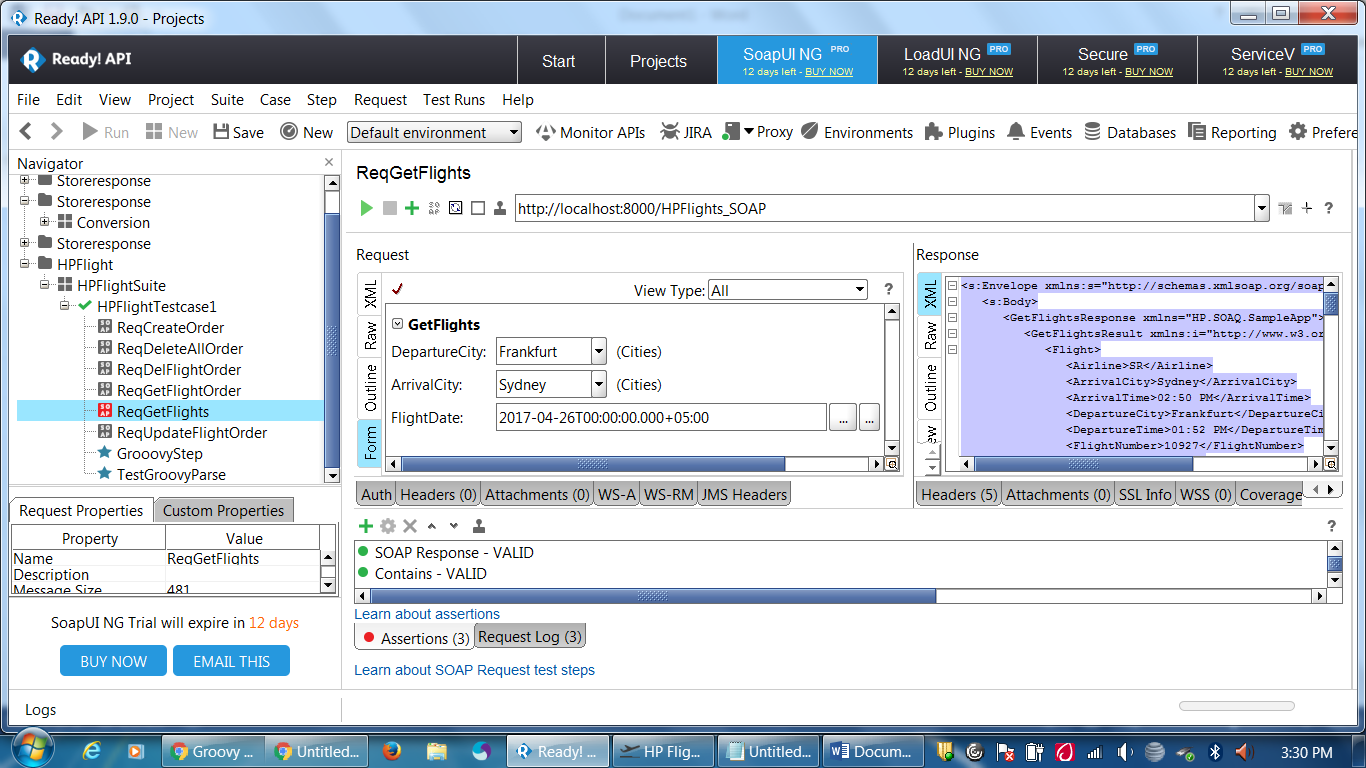
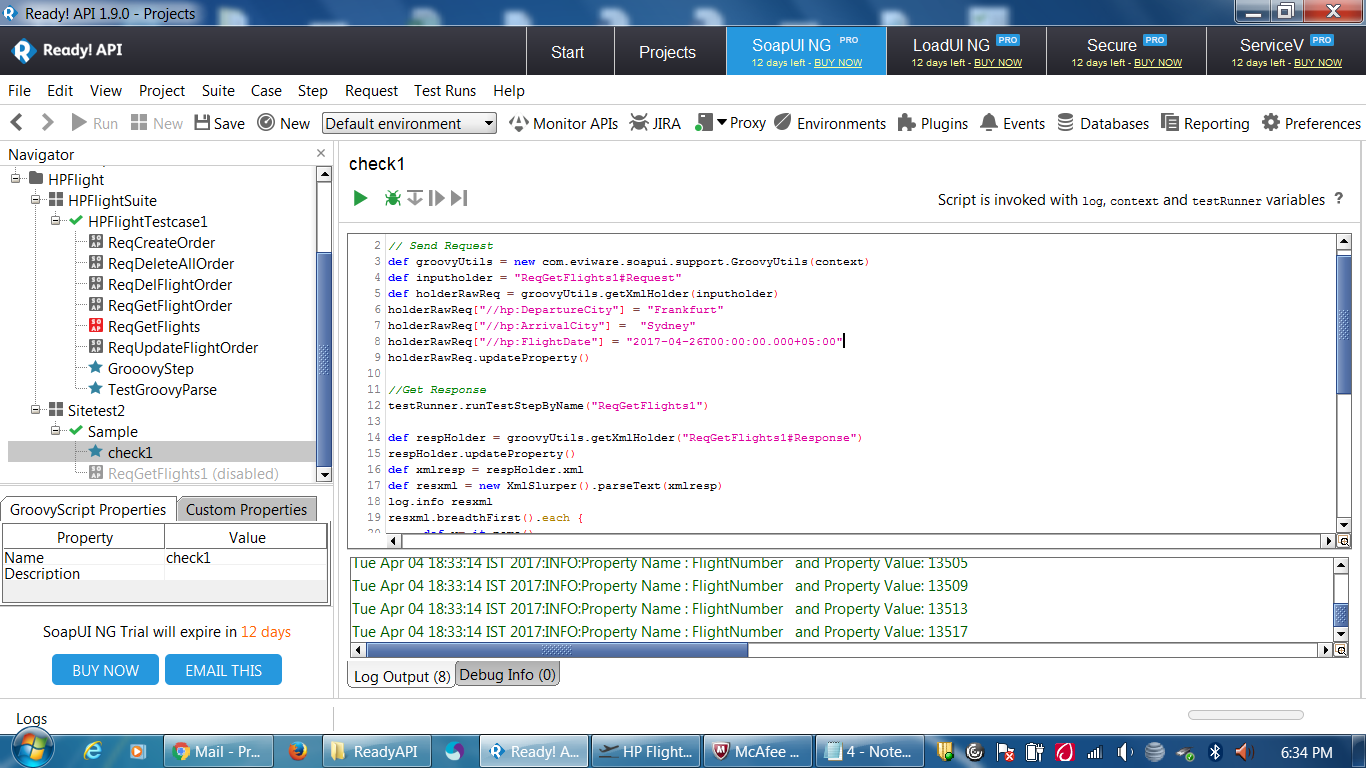
**Method 1: Directly from SOAP UI interface**



**Method 2: GroovyScript : Interface**

|  |
| --- |
| // Send Request  def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)  def inputholder = "ReqGetFlights1#Request"  def holderRawReq = groovyUtils.getXmlHolder(inputholder)  holderRawReq["//hp:DepartureCity"] = "Frankfurt"  holderRawReq["//hp:ArrivalCity"] = "Sydney"  holderRawReq["//hp:FlightDate"] = "2017-04-26T00:00:00.000+05:00"  holderRawReq.updateProperty()  //Get Response  testRunner.runTestStepByName("ReqGetFlights1")  def respHolder = groovyUtils.getXmlHolder("ReqGetFlights1#Response")  respHolder.updateProperty()  def xmlresp = respHolder.xml  def resxml = new XmlSlurper().parseText(xmlresp)  log.info resxml  resxml.breadthFirst().each {  def v= it.name()  def u= it.toString()  //log.info(v)  //log.info(u)  if ( v == "FlightNumber")  {  log.info("Property Name : "+v+ " and Property Value: "+u)  assert u.size() == 5  }        }    } |



**Steps for creating reusable Libraries using Groovy script:**

I've found the need to define a common Library of reusable Groovy Scripts with SOAPUI, there isn't a tutorial or established examples on how this can be achieved so here is an example of how I achieved this.  
  
1) Define a new TestSuite in the root of your project called Library.  
2) Disable the Library TestSuite so that it doesn't get run in an uncontrolled manner.  
3) Define a TestCase under Library, I name this after the module-name.  
4) Define a Groovy Script, I give this the name of the Groovy Class that is going to contain my reusable code.

class Example  
{  
 def log  
 def context  
 def testRunner  
 // Class constructor with same case as Class name  
 def Example(logIn,contextIn,testRunnerIn)  
 {  
 this.log = logIn  
 this.context = contextIn  
 this.testRunner = testRunnerIn  
 }  
}

5) Add the reusable code as method, pass parameters as necessary.

def execute(message)  
 {  
 // do some stuff to prove I've run with right context, etc.  
 log.info testRunner  
 log.info context  
 log.info "return "+message  
 return message  
 }

6) We need to instance of the class; add the following to the end of the Script, outside the class definition. This will place the instance in the project's context.

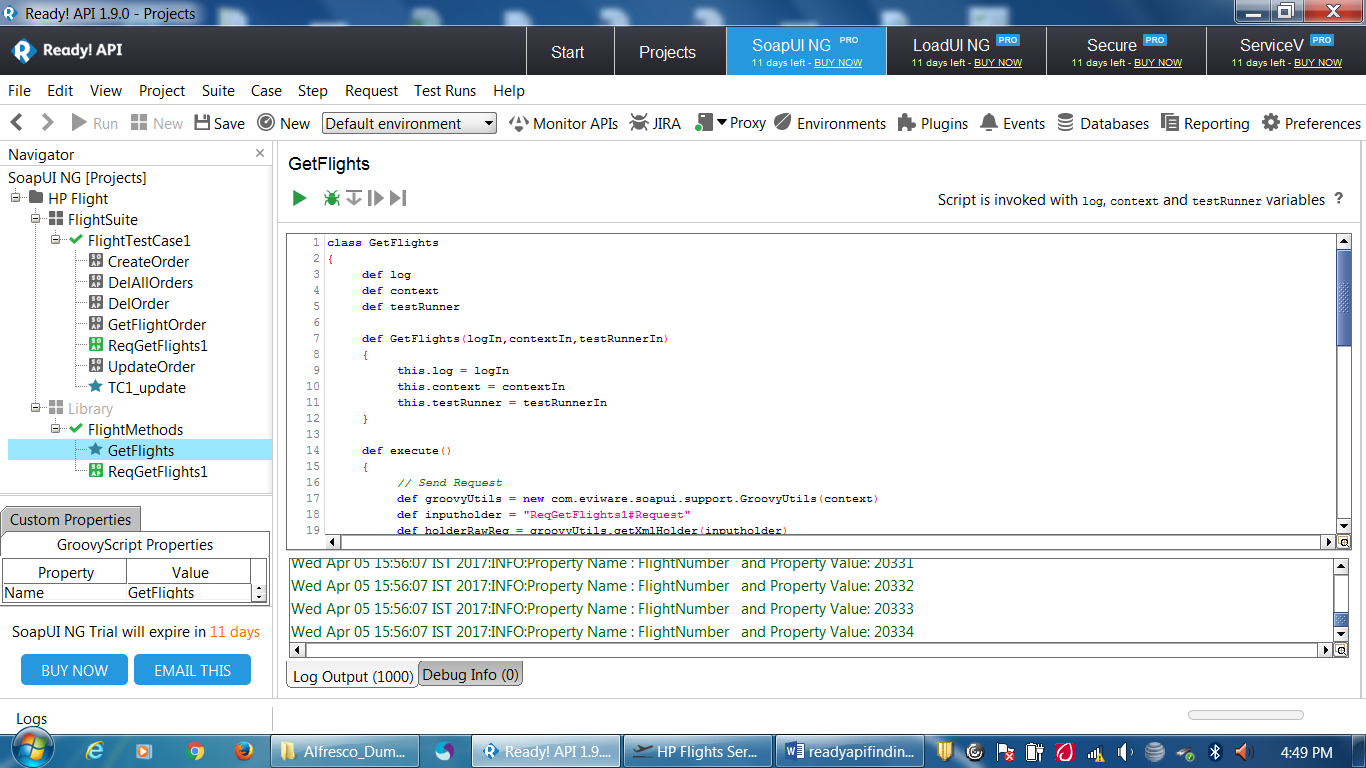
context.setProperty( "example", new Example( log, context, testRunner) )  
log.info "Library Context:"+context

7) You can now reuse the instance in any Groovy Script, with the following Groovy code.

// get a reference to the library TestSuite  
library = testRunner.testCase.testSuite.project.testSuites["Library"]  
  
// find the module within the library   
module = library.testCases["module-name"].testSteps["Example"]  
  
// initialise the library; which places an instance of Example in the context  
module.run(testRunner, context)  
  
// get the instance of example from the context.  
def example = context.example  
  
// run the method, with parameter  
log.info "example.execute() = " + example.execute("Tester")

http://community.smartbear.com/legacyfs/online/smilies/cool.gif Add more modules, classes or methods as necessary.  
9) The instance can be added to the Global context if you want to use it across SOAPUI projects.

|  |
| --- |
| **Implemented:** |



**Groovy Script Module**

|  |
| --- |
| class GetFlights  {  def log  def context  def testRunner  def GetFlights(logIn,contextIn,testRunnerIn)  {  this.log = logIn  this.context = contextIn  this.testRunner = testRunnerIn  }  def execute()  {  // Send Request  def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)  def inputholder = "ReqGetFlights1#Request"  def holderRawReq = groovyUtils.getXmlHolder(inputholder)  holderRawReq["//hp:DepartureCity"] = "Frankfurt"  holderRawReq["//hp:ArrivalCity"] = "Sydney"  holderRawReq["//hp:FlightDate"] = "2017-04-26T00:00:00.000+05:00"  holderRawReq.updateProperty()    //Get Response  testRunner.runTestStepByName("ReqGetFlights1")    def respHolder = groovyUtils.getXmlHolder("ReqGetFlights1#Response")  respHolder.updateProperty()  def xmlresp = respHolder.xml  def resxml = new XmlSlurper().parseText(xmlresp)  log.info resxml  resxml.breadthFirst().each {  def v= it.name()  def u= it.toString()  //log.info(v)  //log.info(u)  if ( v == "FlightNumber")  {  log.info("Property Name : "+v+ " and Property Value: "+u)  // assert u.size() == 5  }  }  return context.values  }  }  context.setProperty("GetFlights", new GetFlights(log,context,testRunner))  def ex = context.GetFlights  ex.execute() |

**Groovy Script Calling this module**

|  |
| --- |
| library = testRunner.testCase.testSuite.project.testSuiteList.get(1)  module = library.testCases["FlightMethods"].testSteps["GetFlights"]  module.run(testRunner,context)  def ex = context.GetFlights  ex.execute() |