SOAP Ui Pro Implementation for Web-service Testing in QA Automation

Wednesday, July 05, 2017 11:38 AM

SOAP UI Pro/ Ready API is a Webservice Testing tool from Smart Bear.

About tool and Key Features

- Tool comes from Smart Bear with a license cost of $499 (increased to $599 effective July 1st 2017)
- It allows tester to test SOAP , REST and MS APIs
- It is user friendly and that is why it was picked in Factset to allow Analyst as well to use it over free tools like Jmeter which are equally or more powerful

Key Feature and comparison

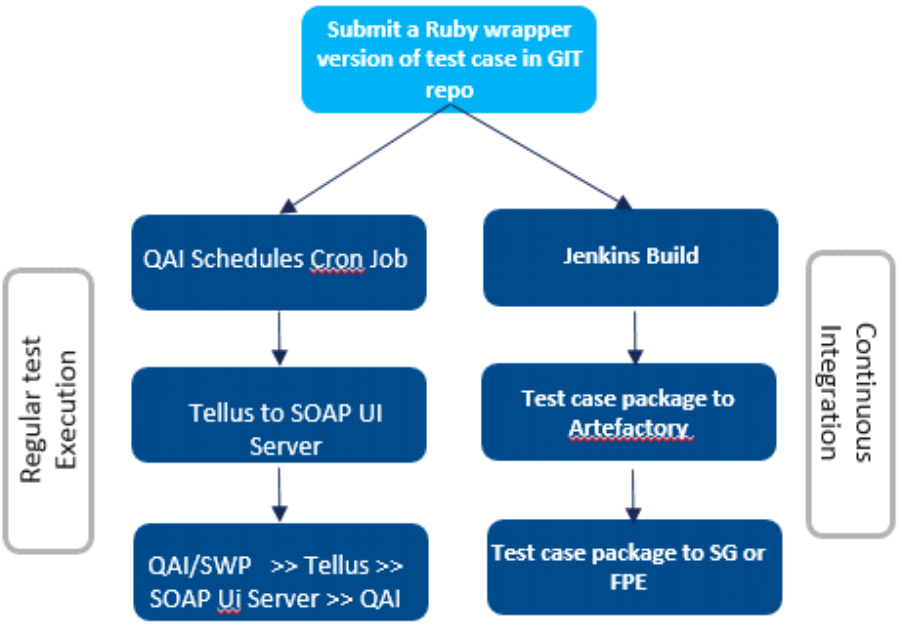
Web-Service Testing tool Comparison

Parameter	SIT	SOAP UI NG Pro	Apache Jmeter
Ease of Use	**Medium**	**Easy**	Complex
Licensing	**In house (On Web)**	▪ Standard License 1 year $499 ▪ Floating licenses available ▪ Free version available **(very limited features)** ▪ **Smart Bear**	**Open Source**
Scalability (Maintenance and Hardware to host)	SIT Server	On User machine and Execution on FPE	On User machine and Execution on FPE
Support	In house (person availability dependent)	Comes with License and very strong support	Good Support
Key features (Built in)	▪ SOAP, REST, MS, DATA Validation, Schema validation, REST Discovery ▪ Multiple Environment support ▪ Hook up with QAI ▪ Scheduling ▪ POST, PUT, GET	▪ SOAP, REST, MS services ▪ **Data Validation and Data driven testing** ▪ Data parameterization using XL, DB, ▪ Support for groovy ▪ Hookup with Tellus and QAI ▪ Assertions ▪ REST Discovery ▪ POST, PUT , GET, Delete etc	▪ **Data Validation and Data driven testing** ▪ REST, MS services ▪ Data parameterization using XL ▪ Hookup with Tellus and QAI ▪ Assertions ▪ SOAP REST Discovery ▪ POST, PUT , GET, Delete etc
Technology Supported	Web-HTTP, HTTPS,SOAP	Web-HTTP, HTTPS,SOAP Database via JDBC,JMS,REST,AMF	Web-HTTP, HTTPS\SOAP Database via JDBC LDAP, Mail-SMTP(S),POP3(S) and IMAP(S), Native commands or shell scripts
Functional Testing	Supported and Primary Function	Supported and Primary Function	Support

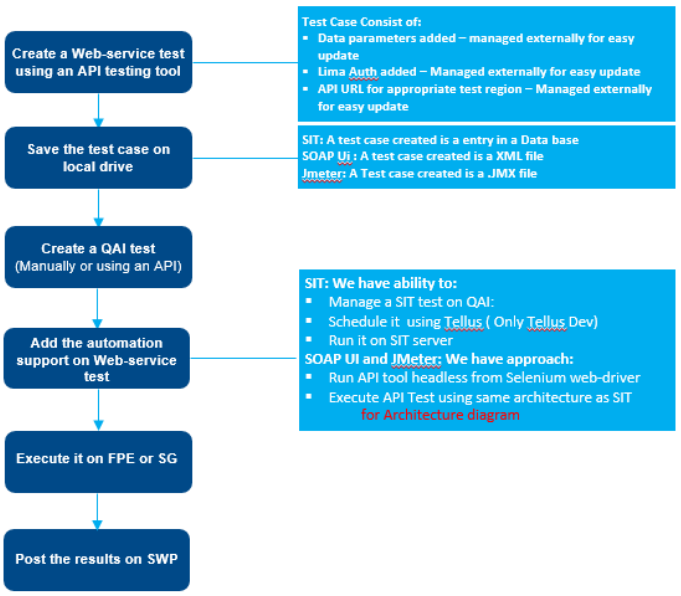
Web-Service Testing tool Comparison

Parameter	SIT	SOAP UI NG Pro	Apache Jmeter
Performance testing	Not Supported	Supported	Supported Primary Function
Integration with Tellus and QAI	QAI is Built Tellus is Built (Tellus Dev)	Should Integrate with Tellus IO, QAI like any-other Selenium test but Ruby support very limited or LIKE SIT	Will be integrated like any Selenium test-case
Headless Execution	No	Yes	Yes
Reporting	Not rich, Support History of runs, at Test case level	Rich reporting	Rich reporting
Service Virtualization	No	Yes	No

Set up Architecture



How Web-service testing fits in our Automation framework?



Changes to Internal FDS tools

QAI Tool – No changes needed
Parent RPD for this effort 29542203
New server (Linux) and set up – Rajul
Tellus changes and updates:
Test Environment – Add SOAP UI – Avanthi to open RPD child to this one 29542203
Project Type – Add SOAP UI
Dispatcher – Arabinda already updated selenium dispatcher for SOAP UI should be able to use it. Avanthi to confirm

DO not want the test cases to be downloaded on tellus server and send it to remote machine. Vasu to work with Ramya.

- Power shell script for test execution on remote machine
- Fall back copy file to selenium ruby test cases

Customization of JEP for SOAP UI test. Following fields will be retained Open RPD as child to 29542203. Avanthi

To keep

- Test Environment
- System
- Product
- Test Plan
- Test cases
- Scheduled Run
- Start
- Email recipient
- VMS User id
- Stage
- HTML reports
- Cassess
- Test Case Location – follow GIT standard for pro
- Ticker list is good to have (Optional at this point, tester should be able use the right ticker list file located on Tellus server or an accessible shared drive

Quick discussion on modularizing the Selenium Ruby wrapper – Ramya. Open an RPD 29542203
Modularize for Test runner.bat
Actual test params
Loggers
Serial # txt file same solution as today
Test case QAI = SOAP UI
Test plan in QAI = Test suite in SOAP UI
Logging at test case level vs suite level – ramya

From <http://is.factset.com/rpd/summary.aspx?messageid=29542203>

WebService Testing_2

Factset - SM - SOA...

SOAP UI Pro Licens...

Master RPD for Productionalizing SOAP UI Pro in Tellus Framework

29542203

From <http://is.factset.com/rpd/summary.aspx?messageId=29542203>

SoapUI best practices

Wednesday, September 20, 2017 11:52 AM

The following sections details the pre-requisites required on Ready! API to get the test cases integrated with our existing Tellus Framework

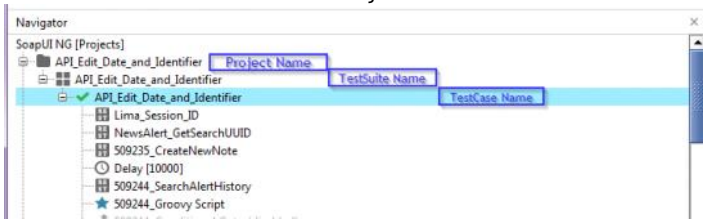
A. Hosting TestCases on Ready! API

On Ready! API, each project hosts just **ONE** test case from QAI. So different test cases from QAI will have to be on different projects on Ready! API

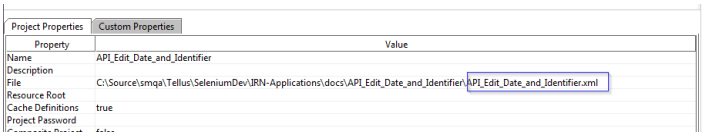
B. Naming Conventions on Ready! API

On Ready! API make sure that the following naming conventions are followed:

1. TestCase Name = TestSuite Name = Project Name

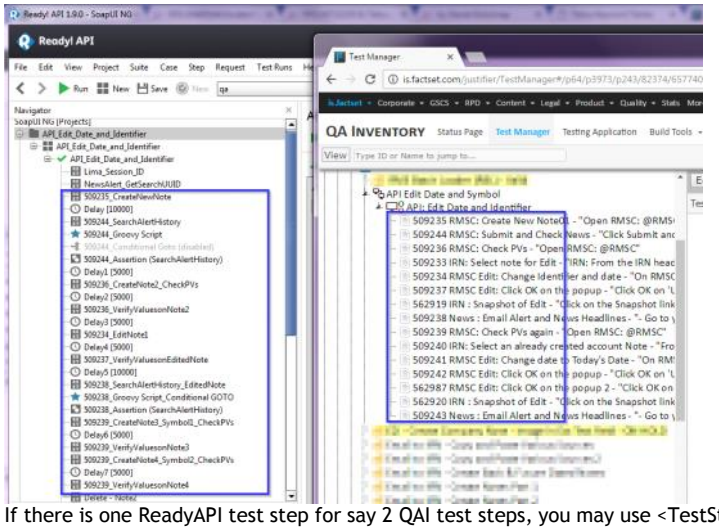


2. Project XML Filename = <TestCase Name>.xml



In the screenshots shown above,
Project Name = TestSuite Name = TestCase Name = **API_Edit_Date_and_Identifier**
Project File name = **API_Edit_Date_and_Identifier.xml**

3. Test Steps: Wherever possible, append the Test Step ID from QAI into the Ready! API test step name – It need not be a 1:1 mapping between QAI and Ready! API test steps, for example, there might be more Ready API! steps to cover a single QAI step etc.

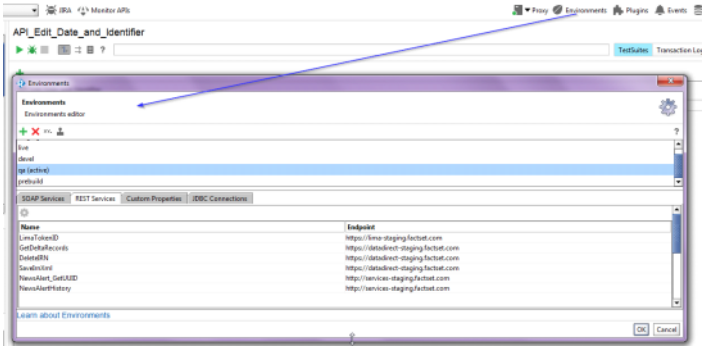


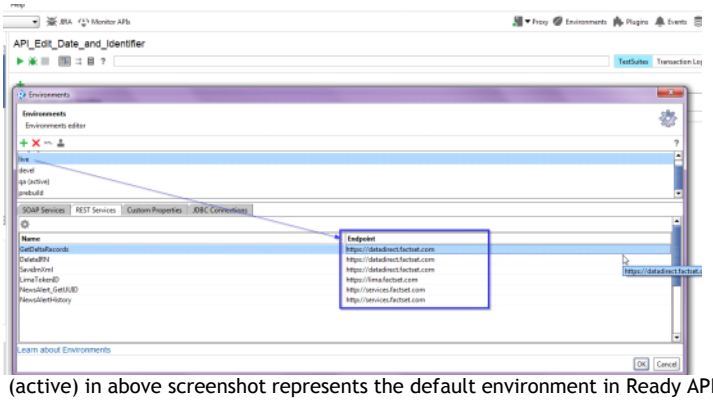
If there is one ReadyAPI test step for say 2 QAI test steps, you may use <TestStepN1>_<TestStepN2>_<Name>

A. Environments on Ready! API

1. Environments on Ready! API should correspond to the stage dropdown values from Tellus JEP – Here are the possible values: **live, released, qa, devel, prebuild**
2. **The Environment Names should all be added in lowercase** – This is very important
3. Add the corresponding end points of your application for different environments

Here are the screenshots showing the different environments and endpoints





Note: (active) in above screenshot represents the default environment in Ready API.

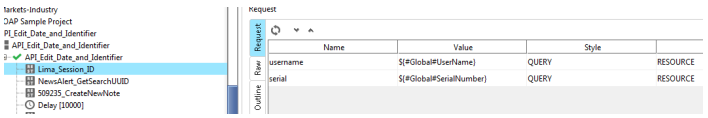
A. Incorporate Global Parameters

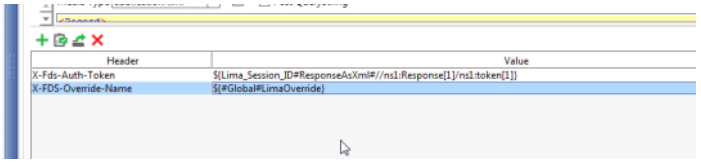
Values from Tellus JEP will be passed as global parameters to your Ready API! test cases

Incorporate global parameters within your Ready! API test cases for the values that you may use from Tellus JEP

Here are some of the examples:

- Replace username FDSQAR_C with ${#Global#UserName}
- Replace serial numbers with ${#Global#SerialNumber}
- Replace Lima Override value for X-FDS-Override-name with ${#Global#LimaOverride}



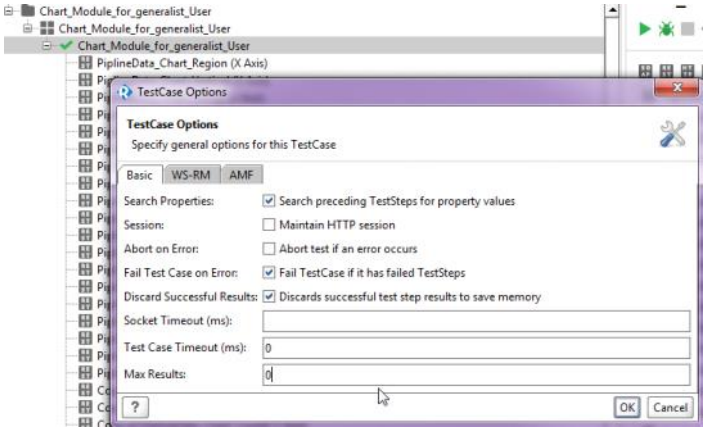


B. maxResults Value at Test Case Level

Ready API! has some default settings for 'Improving Memory Usage' at Test Case Level that may interfere with the way we schedule runs.

On Ready API!, Right Click on your Test Case and select -> Options, you will find that 'Max Results' is set to 5. This setting will interfere with the way we consume the results and hence in some of the tests we are only seeing the results for the last 5 test steps

We need to explicitly make it to **0** (If you specify 0, Ready! API will keep all the test step results in memory.). Please note that once you have changed it and saved your project, the setting should reflect on the xml file as **maxResults="0"**



Please make sure that **maxResults="0"** is reflected in the project xml file too before you check in the script.

C. Assert within Groovy Scripts

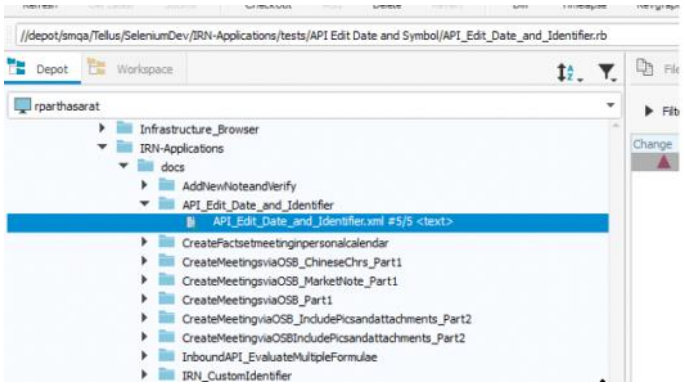
If you have 'Groovy Test Step' within your test case, it is very important that you use 'assert' statements for errors instead of log.error.

Examples:

1. ```
   def fileName = new File("\\\\tellusdev\\output\\RMS-Services\\EvaluateMultipleFqlFormulas.xls")
   assert fileName.exists()
   ```
2. ```
   assert (slurperresponse.contains("responseStatus": "OK"))
   ```
3. ```
   assert pchgVal !="@NA" :"512155:Value under column %Chg is '-' even if values are displayed in %Index Weight Start Date & %Index Weight End Date columns ,Row number "+getRow[b]
   ```
4. ```
   assert electronics_id == "120 Electronics" : "512155: '120 Electronics' is not the first subgroup(2nd row) in the report"
   ```

5. Folder Structure

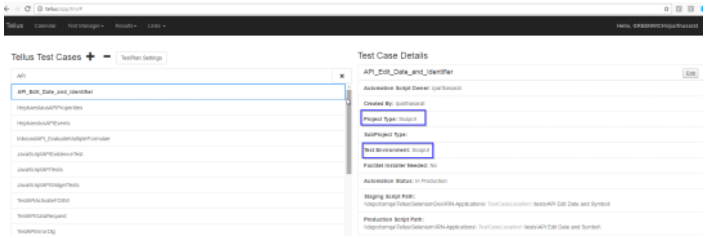
Ready API! XML files are hosted within the docs folder (/docs/<Test case name>/<*.xml>)



6. Tellus Test Manager (TTM) Mappings

To Map the Ready! API test cases on Tellus Test Manager, refer to an example screenshot as below:
Project Type: SoapUI, Test Environment: SoapUI

Please Note: We do not require the corresponding ruby files now for SoapUI tests, but we still need to add the mapping on TTM referencing to a dummy path to the tests folder location like it would have been if you were to submit a ruby file. This is again temporary until we have a proper solution to deal with this change.



Differences between Soap UI PRO and Soap UI Open Source

Tuesday, February 06, 2018 11:04 AM

(From Bhupendar)

Differences between Soap UI PRO and Soap UI Open Source
There is nice comparison done by feature wise on SmartBear website: https://www.soapui.org/professional/soapui-pro/soapui-vs-soapui-pro.html. However, to highlight a few please refer below:

S.NO.	Feature	Comments
1	Multi-Environment Support	This feature allow us to make settings at project level for testing against different environments. At FactSet, we do testing against various stages such as Devel, QA, Live where each stage has different end points to hit. Thus, this is a very vital feature for us.
2	Floating Licenses	This feature comes handy when we want to effectively utilize our licenses amongst scripters such that we have 100% consumption of
3	WSDL	We have some applications such as Code Red, which depends on WSDL services and thus this is nice to have feature.
4	Request/Response Coverage	PRO version gives us JSON view of response data which is easy to read and retrieve data from but the same is not available with free version. Free version provides response in XML format.
5	Test History: Baseline and Comparison	This feature is very vital to recover mistakenly deleted files and compare the existing file with previous version.
6	Coding Free Test Assertion	It allow one to assert based on the response from previous step without having to know any programming knowledge.
7	Data Source Driven	It allows to read data from various sources such as text file, excel, database, etc., which helps us iterate over test data with ease which is not possible with free version.
8	Scripting	PRO version comes with large number of libraries which allow us to write groovy scripts that can interact with UI elements too. With this, if one has requirement to deal with some web component in between

Type of FactSet applications that need Soap UI PRO version

Testing against multiple Environment: If you got an application which needs testing against different environments such as staging, dev, qa, live etc then you need PRO version as it reduces efforts to maintain different scripts for each version. You can do the same stuff with free version but then you'll have to maintain one script each for different stage.

Drivers for Database: If you need connection to different database for retrieving data and then passing it along the web-service call then PRO is the perfect choice as it has support for almost all major database drivers to connect to. One such requirement I see with IS team where they have to connect to JDBC driver which is not available with Open Source version.

Data Source Driven Testing: If we want to read test data from different sources such as XML, spreadsheets, databases etc., then PRO is best choice for us as it can read data from various

sources which helps us to test application against different data sets in each iteration. Open source version doesn't have this feature and thus for each data set one has to write same steps again which creates lot of redundant work.

Scripting Libraries: When you need to run some external script to initiate some action which is part of your web-service testing then you need external libraries to achieve it. Though groovy scripting is available in free version of Soap UI, it comes with very limited libraries with which it is not possible to achieve complex functionalities. For example, we had one of the use case where only after performing a click on UI component the web-service request would proceed further and we used Selenium-WebDriver + PhanthomJS to implement this functionality.

Use of Test Complete Utilities: If you got a use case wherein you want to call the utility functions written in Test Complete, you can do so with PRO version of Soap UI. This will reduce your efforts to rewrite utilities specifically in groovy script again. Platform team has used this feature earlier to use excel charts utilities.

Additional Features which increases our speed of automation are:

1. **Point and Click:** This enable us to simply right-click and assertions on the response data. It also allows us to get data from previous steps without having to write any code and thus enables non-programmer to easily automate API testing. This is one of the heavily used feature by teams. Not to mention, this feature works in groovy script editor too !!
2. **Huge Assertion Library:** Assertions are key to verify the outcome of request and it is not possible to just rely on basic message assertions. PRO version provides option to write Regular Expressions, Wild Cards etc., while writing assertions. This is also one of the heavily used feature by teams.
3. **Plugin Manager:** PRO version comes with an option to install many plugins which can be used to integrate with various other tools such as JIRA, Jenkins, GitHub etc., which will help us achieve CID for Soap UI projects.
4. Security Testing (Limited version)
5. Load Testing (Limited version)
6. Service Virtualization (Limited version)