Import Execution Results

- Supported Import Formats
  - Cucumber JSON output format
  - JUnit XML output format
  - NUnit v3.0 XML output format
  - NUnit v2.6 XML output format
  - Xray JSON format
    - Example 1: Automated Tests
    - Example 2: Manual Tests
    - Example 3: Generic Test
- Importing Multiple Execution results
  - Xamarin Test Cloud

External execution results from either automated and manual Tests can be imported to Jira. This operation may be done in one of two ways:

- **Manually**, using the *Import Execution Results* action accessible from the Test Execution issue screen
- **via the REST API** to integrate with Continuous Integration (CI) platforms or other external execution processes. Please refer to the Xray REST API.

When manually importing execution results, the current Test Execution issue will be updated with the results. When using the REST API, you can specify an existing Test Execution issue in Jira; if the Test Execution key is missing, Xray will create a new Test Execution issue based on the information provided.

Supported Import Formats

Xray supports the following formats for importing execution results:

1. Cucumber JSON output format
2. JUnit XML output format
3. NUnit XML output format
4. Xray JSON format

Cucumber JSON output format

The Cucumber tool is capable of generating multiple reports for an execution. In order to import the execution results to Xray, Cucumber must generate a JSON output (example here) using the following arguments:

```
-f, --format FORMAT       How to format features. Available formats compatible with Xray for JIRA:
                     json        : Prints the feature as JSON
                json_pretty : Prints the feature as prettified JSON
-x, --expand              Expand Scenario Outline Tables in output.

Ex:
> cucumber -x -f json
```

JUnit XML output format

Xray supports a JUnit XML format for importing execution results.

```
<?xml version="1.0" encoding="UTF-8" ?>
<testsuite tests="15" failures="0" name="ut.com.xpandit.raven.service.impl.IssueDataSetTest" time="0.163"
errors="0" skipped="0">
  <properties>
    ...
  </properties>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidLimitOverflowOption_returnsExpectedSubset" time="0.114"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withNullOptionsAndValidIssue_throwsIllegalArgumentException" time="0.002"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidEmptyOptions_returnsAllIssues" time="0.002"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidGlobalSearchOptions_returnsExpectedTests" time="0.016"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndInvalidColumnSearchOption_returnsAllTests" time="0.007"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidLimitUnderOption_returnsExpectedSubset" time="0.001"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidGlobalSearchOptionThatMachesIssueKey_returnsExpectedTestWithMatchedKey"
time="0.006"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidSummaryColumnAscSortOption_returnsExpectedIssuesInAscOrder" time="0.006"
/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidSummaryColumnDescSortOption_returnsExpectedIssuesInDescOrder" time="
0.002"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidGlobalSearchOptionThatMatchesAllElements_returnsAllTests" time="0.001"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidColumnSearchOptionThatMatchesOneElement_returnsOneTest" time="0.002"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidColumnSearchOptionThatMatchesNoIssue_returnsEmptyList" time="0.001"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidGlobalSearchOptionThatMachesNoIssue_returnsEmptyList" time="0.001"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidKeyColumnDescSortOption_returnsExpectedIssuesInDescOrder" time="0.001"/>
  <testcase classname="ut.com.xpandit.raven.service.impl.IssueDataSetTest" name="
testApplyOptions_withValidIssueAndValidKeyColumnAscSortOption_returnsExpectedIssuesInAscOrder" time="0.001"/>
</testsuite>
```

NUnit v3.0 XML output format

Xray supports an NUnit v3.0 XML format for importing execution results.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<test-run id="0" testcasecount="14" total="14" passed="13" failed="1" inconclusive="0" skipped="0" asserts="14"
result="Failed" portable-engine-version="3.3.0.0" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:
03Z" duration="0.140400">
<test-suite type="Assembly" id="1021" name="x, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" fullname="
x, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null" runstate="Runnable" testcasecount="14" result="
Failed" site="Child" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.110549"
total="14" passed="13" failed="1" inconclusive="0" skipped="0" asserts="14">
  <settings>
        <setting name="WorkDirectory" value="C:\Users\Sergio\x" />
  </settings>
  <failure>
        <message><![CDATA[One or more child tests had errors]]></message>
  </failure>
  <test-suite type="TestFixture" id="1000" name="TestClass" fullname="TestClass" classname="TestClass"
runstate="Runnable" testcasecount="2" result="Failed" site="Child" start-time="2016-12-26 14:36:03Z" end-time="
2016-12-26 14:36:03Z" duration="0.084668" total="2" passed="1" failed="1" inconclusive="0" skipped="0" asserts="
2">
        <failure>
          <message><![CDATA[One or more child tests had errors]]></message>
        </failure>
        <test-suite type="ParameterizedMethod" id="1003" name="SubtractTest" fullname="TestClass.SubtractTest"
```

```
classname="TestClass" runstate="Runnable" testcasecount="2" result="Failed" site="Child" start-time="2016-12-26
14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.080887" total="2" passed="1" failed="1" inconclusive="0"
skipped="0" asserts="2">
                <properties>
                        <property name="Requirement" value="DEV-771" />
                  </properties><failure>
                <message><![CDATA[One or more child tests had errors]]></message>
          </failure>
          <test-case id="1001" name="SubtractTest(1)" fullname="TestClass.SubtractTest(1)" methodname="
SubtractTest" classname="TestClass" runstate="Runnable" seed="1166833138" result="Failed" start-time="2016-12-
26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.043525" asserts="1">
                <failure>
                  <message><![CDATA[  Expected: 10
But was:  1
]]></message>
                  <stack-trace><![CDATA[at TestClass.SubtractTest(Int32 x) in C:\Users\Sergio\x\TestClass.cs:
line 13
]]></stack-trace>
                </failure>
          </test-case>
          <test-case id="1002" name="SubtractTest(10)" fullname="TestClass.SubtractTest(10)" methodname="
SubtractTest" classname="TestClass" runstate="Runnable" seed="1003146807" result="Failed" start-time="2016-12-
26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="12.000098" asserts="1" />
        </test-suite>
  </test-suite>
  <test-suite type="TestSuite" id="1022" name="x" fullname="x" runstate="Runnable" testcasecount="12" result="
Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.015218" total="12"
passed="12" failed="0" inconclusive="0" skipped="0" asserts="12">
        <test-suite type="TestFixture" id="1004" name="CalculatorTests" fullname="x.CalculatorTests" classname="
x.CalculatorTests" runstate="Runnable" testcasecount="12" result="Passed" start-time="2016-12-26 14:36:03Z" end-
time="2016-12-26 14:36:03Z" duration="0.014979" total="12" passed="12" failed="0" inconclusive="0" skipped="0"
asserts="12">
          <test-suite type="ParameterizedMethod" id="1008" name="CanAddNumbers" fullname="x.CalculatorTests.
CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="
2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.004228" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
                  <properties>
            <property name="Requirement" value="DEV-771" />
          </properties>
                <test-case id="1005" name="CanAddNumbers(1,1,2)" fullname="x.CalculatorTests.CanAddNumbers
(1,1,2)" methodname="CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" seed="1846389584" result="
Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.001194" asserts="1" />
                <test-case id="1006" name="CanAddNumbers(-1,-1,-2)" fullname="x.CalculatorTests.CanAddNumbers
(-1,-1,-2)" methodname="CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" seed="1113780989"
result="Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000067" asserts="
1" />
                <test-case id="1007" name="CanAddNumbers(100,5,105)" fullname="x.CalculatorTests.CanAddNumbers
(100,5,105)" methodname="CanAddNumbers" classname="x.CalculatorTests" runstate="Runnable" seed="1585332966"
result="Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000103" asserts="
1" />
          </test-suite>
          <test-suite type="ParameterizedMethod" id="1020" name="CanDivide" fullname="x.CalculatorTests.
CanDivide" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="2016-
12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.004041" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
                  <properties>
            <property name="Requirement" value="DEV-771" />
          </properties>
                <test-case id="1017" name="CanDivide(1,1,1)" fullname="x.CalculatorTests.CanDivide(1,1,1)"
methodname="CanDivide" classname="x.CalculatorTests" runstate="Runnable" seed="1285501252" result="Passed"
start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000354" asserts="1" />
                <test-case id="1018" name="CanDivide(-1,-1,1)" fullname="x.CalculatorTests.CanDivide(-1,-1,1)"
methodname="CanDivide" classname="x.CalculatorTests" runstate="Runnable" seed="1436436719" result="Passed"
start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000073" asserts="1" />
                <test-case id="1019" name="CanDivide(100,5,20)" fullname="x.CalculatorTests.CanDivide
(100,5,20)" methodname="CanDivide" classname="x.CalculatorTests" runstate="Runnable" seed="213310888" result="
Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000060" asserts="1" />
          </test-suite>
          <test-suite type="ParameterizedMethod" id="1016" name="CanMultiply" fullname="x.CalculatorTests.
CanMultiply" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="
2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.002759" total="3" passed="3" failed="0"
```

```
inconclusive="0" skipped="0" asserts="3">
                <test-case id="1013" name="CanMultiply(1,1,1)" fullname="x.CalculatorTests.CanMultiply(1,1,1)"
methodname="CanMultiply" classname="x.CalculatorTests" runstate="Runnable" seed="1192735127" result="Passed"
start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000331" asserts="1">
                                <properties>
            <property name="label" value="multiplication" />
          </properties>
                </test-case>
                <test-case id="1014" name="CanMultiply(-1,-1,1)" fullname="x.CalculatorTests.CanMultiply(-1,
-1,1)" methodname="CanMultiply" classname="x.CalculatorTests" runstate="Runnable" seed="39988064" result="
Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000059" asserts="1">
                                <properties>
            <property name="label" value="multiplication" />
          </properties>
                </test-case>
                <test-case id="1015" name="CanMultiply(100,5,500)" fullname="x.CalculatorTests.CanMultiply
(100,5,500)" methodname="CanMultiplyAgain" classname="x.CalculatorTests" runstate="Runnable" seed="1462346243"
result="Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000052" asserts="
1">
                                <properties>
            <property name="requirement" value="DEV-34" />
          </properties>
                </test-case>
          </test-suite>
          <test-suite type="ParameterizedMethod" id="1012" name="CanSubtract" fullname="x.CalculatorTests.
CanSubtract" classname="x.CalculatorTests" runstate="Runnable" testcasecount="3" result="Passed" start-time="
2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.002827" total="3" passed="3" failed="0"
inconclusive="0" skipped="0" asserts="3">
                <properties>
            <property name="requirement" value="DEV-328" />
          </properties>
                <test-case id="1009" name="CanSubtract(1,1,0)" fullname="x.CalculatorTests.CanSubtract(1,1,0)"
methodname="CanSubtract" classname="x.CalculatorTests" runstate="Runnable" seed="1019357734" result="Failed"
start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000303" asserts="1">
                  <failure>
                        <message><![CDATA[Error subtracting]]></message>
                  </failure>
          </test-case>
                <test-case id="1010" name="CanSubtract(-1,-1,0)" fullname="x.CalculatorTests.CanSubtract(-1,
-1,0)" methodname="CanSubtract" classname="x.CalculatorTests" runstate="Runnable" seed="1322022615" result="
Failed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000056" asserts="1" >
                           <failure>
                <message><![CDATA[Error subtracting]]></message>
          </failure>
          </test-case>
                <test-case id="1011" name="CanSubtract(100,5,95)" fullname="x.CalculatorTests.CanSubtract
(100,5,95)" methodname="CanSubtract" classname="x.CalculatorTests" runstate="Runnable" seed="4493553" result="
Passed" start-time="2016-12-26 14:36:03Z" end-time="2016-12-26 14:36:03Z" duration="0.000053" asserts="1" />
          </test-suite>
        </test-suite>
  </test-suite>
</test-suite>
</test-run>
```

NUnit v2.6 XML output format

Xray supports an NUnit v2.6 XML format for importing execution results.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!--This file represents the results of running a test suite-->
<test-results name="D:\Work\XpandIT\Repo\jm.customermobility\JMMobilityServices\JMMobilityServices.
ShoppingLists.Tests\bin\Debug\JMMobilityServices.ShoppingLists.Tests.dll" total="17" errors="0" failures="0"
not-run="0" inconclusive="0" ignored="0" skipped="0" invalid="0" date="2017-01-18" time="15:46:53">
  <environment nunit-version="2.6.4.14350" clr-version="2.0.50727.8745" os-version="Microsoft Windows NT
6.2.9200.0" platform="Win32NT" cwd="D:\Work\Apps\NUnit-2.6.4\bin" machine-name="LAPTOP-XPRT" user="rui.tome"
user-domain="LAPTOP-XPRT" />
  <culture-info current-culture="pt-PT" current-uiculture="en-GB" />
  <test-suite type="Assembly" name="D:\Work\XpandIT\Repo\jm.
customermobility\JMMobilityServices\JMMobilityServices.ShoppingLists.Tests\bin\Debug\JMMobilityServices.
```

```
ShoppingLists.Tests.dll" executed="True" result="Success" success="True" time="59.109" asserts="0">
    <results>
      <test-suite type="Namespace" name="JMMobilityServices" executed="True" result="Success" success="True"
time="59.104" asserts="0">
        <results>
          <test-suite type="Namespace" name="ShoppingLists" executed="True" result="Success" success="True"
time="59.104" asserts="0">
            <results>
              <test-suite type="Namespace" name="Tests" executed="True" result="Success" success="True" time="
59.103" asserts="0">
                <results>
                  <test-suite type="Namespace" name="Controllers" executed="True" result="Success" success="
True" time="59.103" asserts="0">
                    <results>
                      <test-suite type="TestFixture" name="ShoppingListsControllerTests" executed="True"
result="Success" success="True" time="59.102" asserts="0">
                        <results>
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.AddClient" executed="True" result="Success" success="True" time="6.428" asserts="
4" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.AddShoppingListItem" executed="True" result="Success" success="True" time="2.837"
asserts="1" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.CreateShoppingList" executed="True" result="Success" success="True" time="4.765"
asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.DeleteShoppingList" executed="True" result="Success" success="True" time="4.657"
asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetPublicShoppingLists" executed="True" result="Success" success="True" time="
2.829" asserts="4" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetShoppingList" executed="True" result="Success" success="True" time="1.734"
asserts="1" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetShoppingLists" executed="True" result="Success" success="True" time="1.522"
asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.GetSuggestedProducts" executed="True" result="Success" success="True" time="0.398"
asserts="0" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveAllShoppingListItems" executed="True" result="Success" success="True" time="
4.373" asserts="4" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveBoughtShoppingListItems" executed="True" result="Success" success="True"
time="4.575" asserts="5" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveClient" executed="True" result="Success" success="True" time="3.476"
asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.RemoveShoppingListItem" executed="True" result="Success" success="True" time="
4.118" asserts="2" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.TestAuthorization" executed="True" result="Success" success="True" time="0.355"
asserts="1">
                            <properties>
                              <property name="Requirement" value="XPTO-15" />
                            </properties>
                          </test-case>
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UnmarkBoughtShoppingListItems" executed="True" result="Success" success="True"
time="4.145" asserts="5" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UpdateShoppingListItemBoughtStatus" executed="True" result="Success" success="
True" time="3.972" asserts="3" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UpdateShoppingListItemQuantity" executed="True" result="Success" success="True"
time="3.560" asserts="3" />
                          <test-case name="JMMobilityServices.ShoppingLists.Tests.Controllers.
ShoppingListsControllerTests.UpdateShoppingListName" executed="True" result="Success" success="True" time="
```

```
2.927" asserts="2" />
                        </results>
                      </test-suite>
                    </results>
                  </test-suite>
                </results>
              </test-suite>
            </results>
          </test-suite>
        </results>
      </test-suite>
    </results>
  </test-suite>
</test-results>
```

Xray JSON format

Xray also supports a custom and generic format for importing execution results. This allows you to import any execution result as long as it is in JSON format. This way, you can import any execution result to Jira, either from automated or manual Tests.

Importing Execution Evidences

This format supports importing execution evidences (attachments) embedded in the JSON file encoded in **Base 64**.

Creating new Test Execution issues

The **info** object allows you to specify specific execution information when creating new Test Execution issues.

Importing results to Manual Tests

You can import manual Test execution results using the **steps** element for specifying the Test step results.

The JSON results file must follow this schema:

```
{
    "type": "object",
    "$schema": "http://json-schema.org/draft-03/schema",
    "required": false,
    "properties": {
        "testExecutionKey": {
            "type": "string",
            "required": false
        },
        "info": {
            "type": "object",
            "required": false,
            "properties": {
                "project": {
                    "type": "string",
                    "required": false
                },
                "summary": {
                    "type": "string",
                    "required": true
                },
                "description": {
                    "type": "string",
                    "required": false
                },
                "user": {
                    "type": "string",
                    "required": false
                },
                "version": {
```

```
            "type": "string",
            "required": false
        },
        "revision": {
            "type": "string",
            "required": false
        },
        "startDate": {
            "type": "string",
            "required": false
        },
        "finishDate": {
            "type": "string",
            "required": false
        }
    }
},
"tests": {
    "type": "array",
    "required": true,
    "items": {
        "type": "object",
        "required": false,
        "properties": {
            "comment": {
                "type": "string",
                "required": false
            },
            "finish": {
                "type": "string",
                "required": false
            },
            "start": {
                "type": "string",
                "required": false
            },
            "status": {
                "type": "string",
                "required": true
            },
            "testKey": {
                "type": "string",
                "required": true
            },
                                "executedBy": {
                "type": "string",
                "required": false
            },
            "evidences": {
                "type": "array",
                "required": false,
                "items": {
                    "type": "object",
                    "required": false,
                    "properties": {
                        "data": {
                            "type": "string",
                            "required": true
                        },
                        "filename": {
                            "type": "string",
                            "required": true
                        },
                        "contentType": {
                            "type": "string",
                            "required": false
                        }
                    }
                }
            },
            "results": {
```

```
        "type": "array",
        "required": false,
        "items": {
            "type": "object",
            "required": false,
            "properties": {
                "name": {
                    "type": "string",
                    "required": true
                },
                "duration": {
                    "type": "long",
                    "required": false
                },
                "log": {
                    "type": "string",
                    "required": false
                },
                "status": {
                    "type": "string",
                    "required": true
                },
                "examples": {
                    "type": "array",
                    "required": false,
                    "items": {
                        "type": "string"
                    }
                }
            }
        }
    },
    "examples": {
        "type": "array",
        "required": false,
        "items": {
            "type": "string"
        }
    },
    "steps": {
        "type": "array",
        "required": false,
        "items": {
            "type": "object",
            "required": false,
            "properties": {
                "status": {
                    "type": "string",
                    "required": true
                },
                "comment": {
                    "type": "string",
                    "required": false
                },
                "evidences": {
                    "type": "array",
                    "required": false,
                    "items": {
                        "type": "object",
                        "required": false,
                        "properties": {
                            "data": {
                                "type": "string",
                                "required": true
                            },
                            "filename": {
                                "type": "string",
                                "required": true
                            },
                            "contentType": {
                                "type": "string",
```

```
                                        "required": false
                                    }
                                }
                            }
                        },
                                                    "actualResult": {
                                                            "type": "string",
                                                            "required": false
                                                    }
                    }
                }
            }
        }
    }
}
```

Example 1: Automated Tests

This is an example of a JSON file with execution results that will update Automated Tests in Jira. The last item DEMO-9 must be a Cucumber Test because the results contain examples.

```
{
    "tests" : [
        {
            "testKey" : "DEMO-7",
            "start" : "2013-05-03T11:47:35+01:00",
            "finish" : "2013-05-03T11:50:56+01:00",
            "comment" : "Test was OK but the performance is very poor",
            "status" : "PASS"
        },
        {
            "testKey" : "DEMO-8",
            "start" : "2013-05-03T12:14:12+01:00",
            "finish" : "2013-05-03T12:15:23+01:00",
            "comment" : "Performance was better this time, in the context of test set DEMO-10.",
            "status" : "PASS"
        },
        {
            "testKey" : "DEMO-9",
            "start" : "2013-05-03T12:19:23+01:00",
            "finish" : "2013-05-03T12:20:01+01:00",
            "comment" : "Error decreasing space shuttle speed.",
            "status" : "FAIL",
                        "examples" : [
                                "PASS",
                                "PASS",
                                "PASS",
                                "PASS",
                                "PASS",
                                "FAIL"
                        ]
        }
    ]
}
```

Example 2: Manual Tests

This is an example of a JSON file with execution results that will update a Manual Test in Jira.

```
{
    "tests" : [
        {
            "testKey" : "DEMO-57",
            "start" : "2014-08-30T12:19:23+01:00",
            "finish" : "2014-08-30T12:20:01+01:00",
            "comment" : "Error executing step 2!",
            "status" : "FAIL",
                        "steps": [
                                {
                    "status": "PASS",
                    "comment": "Step 1: OK"
                },
                {
                    "status": "FAIL",
                    "comment": "Step 2 *Failed* with an unexpected error message",
                    "evidences" : [
                        {
                            "data": "(... base 64 encoded ...)",
                            "filename": "screenshot1.jpg",
                            "contentType": "image/jpeg"
                        }
                    ]
                }
                        ]
        }
    ]
}
```

Example 3: Generic Test

This is an example of a JSON file with execution results that will update a Generic Test in Jira.

```
{
    "tests" : [
        {
            "testKey" : "ABC-129",
            "start" : "2014-08-30T11:47:35+01:00",
            "finish" : "2014-08-30T11:50:56+01:00",
            "comment" : "Successful execution",
            "status" : "PASS"
        }
    ]
}
```

Importing Multiple Execution results

Xray supports importing multiple results for the same Test issues in the same execution. These results often indicate different contexts/environments where the same Test must be executed. Xray will group all executions of the same Test in a single Test Run and present all execution information, including the different contexts in the Execution page.

When importing execution results using the Cucumber JSON output formatter, multiple executions for the same scenario/Test issue will be merged into a single Test Run.

When importing execution results using the Xray JSON format, the **results** JSON element must contain all execution results for the same Test Run.

Xamarin Test Cloud

If you are using the Xamarin Test Cloud for executing **Cucumber** mobile Tests in different combinations of mobile devices and operating systems, you can import the results to Jira by making a compressed zip file containing the multiple Cucumber JSON files.

Learn more

Please see Testing using Calabash and Xamarin Test Cloud in Ruby and the "bundle" endpoint in the REST API.