Import Execution Results - REST v2

- Importing Execution Results
  - Xray JSON results
  - Xray JSON results Multipart
  - Cucumber JSON results
  - Cucumber JSON results Multipart
  - JUnit XML results
  - JUnit XML results Multipart
  - NUnit XML results
  - NUnit XML results Multipart
  - xUnit XML results
  - xUnit XML results Multipart
  - TestNG XML results
  - TestNG XML results Multipart
  - Robot Framework XML results
  - Robot Framework XML results Multipart
  - Behave JSON results
  - Behave JSON results Multipart

Importing Execution Results

Execution results can be imported to Jira through JSON/XML representation formats specified in Integrating with Testing Frameworks or using the proprietary Xray JSON format.

For each import file format, Xray provides a specific REST endpoint:

Xray JSON format	**/api/v2/import/execution**
Xray JSON format Multipart	**/api/v2/import/execution/multipart**
Cucumber JSON output format	**/api/v2/import/execution/cucumber**
Cucumber JSON output format Multipart	**/api/v2/import/execution/cucumber/multipart**
JUnit XML output format	**/api/v2/import/execution/junit**
JUnit XML output format Multipart	**/api/v2/import/execution/junit/multipart**
NUnit XML output format	**/api/v2/import/execution/nunit**
NUnit XML output format Multipart	**/api/v2/import/execution/nunit/multipart**
xUnit XML output format	**/api/v2/import/execution/xunit**
xUnit XML output format Multipart	**/api/v2/import/execution/xunit/multipart**
TestNG XML output format	**/api/v2/import/execution/testng**
TestNG XML output format Multipart	**/api/v2/import/execution/testng/multipart**
Robot Framework XML output format	**/api/v2/import/execution/robot**
Robot Framework XML output format multipart	**/api/v2/import/execution/robot/multipart**

Xray JSON results

When importing execution results using Xray JSON result format in a Continuous Integration environment, you can specify which Test Execution issue to import the results on using the **testExecutionKey** property. Alternatively, you can create a new Test Execution for the execution results and specify the Test Execution issue fields in the **info** object.

Import the execution results present in query variable "***executionResults***".

Request

Example 1: new Test Execution

Example Input

```
{
    "info" : {
        "summary" : "Execution of automated tests for release v1.3",
                "description" : "This execution is automatically created when importing execution results
from an external source",
                "version" : "v1.3",
                "user" : "admin",
                "revision" : "1.0.42134",
                "startDate" : "2014-08-30T11:47:35+01:00",
                "finishDate" : "2014-08-30T11:53:00+01:00",
                "testPlanKey" : "DEMO-100",
                "testEnvironments": ["iOS", "Android"]
    },
    "tests" : [
        {
            "testKey" : "DEMO-6",
            "start" : "2014-08-30T11:47:35+01:00",
            "finish" : "2014-08-30T11:50:56+01:00",
            "actualResult" : "Successful execution",
            "status" : "PASSED"
        },
        {
            "testKey" : "DEMO-7",
            "start" : "2014-08-30T11:51:00+01:00",
            "finish" : "2014-08-30T11:52:30+01:00",
            "actualResult" : "Execution failed. Example #5 FAIL.",
            "status" : "FAILED",
                        "evidence" : [
                    {
                        "data":
"iVBORw0KGgoAAAANSUhEUgAABkIAAAO9CAYAAADezXv6AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEn(...base64
file enconding)",
                                        "filename": "image21.jpg",
                    "contentType": "image/jpeg"
                }
                        ],
            "examples" : [
                "PASSED",
                "PASSED",
                "PASSED",
                "PASSED",
                "FAILED"
            ],
                        "steps": [
                {
                    "status": "PASSED",
                    "comment": "Comment on Test Step Result 1",
                                        "actualResult" : "Step Passed with Status Code 200",
                    "evidences" : [
                        {
                            "data":
"iVBORw0KGgoAAAANSUhEUgAABkIAAAO9CAYAAADezXv6AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEn(...base64
file enconding)",
                                                                "filename": "image22.jpg",
                        "contentType": "image/jpeg"
                        }
                                        ]
                                    }
                        ],
                    "defects" : [
                                        "DEMO-10",
                                        "DEMO-11"
                ]
        }
    ]
}
```

Example 2: update existing Test Execution

Example Input

```
{
    "testExecutionKey": "DEMO-1206",
    "info" : {
        "summary" : "Execution of automated tests for release v1.3",
                "description" : "This execution is automatically created when importing execution results
from an external source",
                "version" : "v1.3",
                "user" : "admin",
                "revision" : "1.0.42134",
                "startDate" : "2014-08-30T11:47:35+01:00",
                "finishDate" : "2014-08-30T11:53:00+01:00",
                "testPlanKey" : "DEMO-100",
                "testEnvironments": ["iOS", "Android"]
    },
    "tests" : [
        {
            "testKey" : "DEMO-6",
            "start" : "2014-08-30T11:47:35+01:00",
            "finish" : "2014-08-30T11:50:56+01:00",
            "comment" : "Successful execution",
            "status" : "PASSED"
        }
     ]
}
```

Example Request

curl -H "Content-Type: application/json" -X POST -H "Authorization: Bearer $token" --data @"data.json" https://xray.cloud.getxray.app/api/v2/import/execution

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** No execution results were provided**.**

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

Xray JSON results Multipart

Xray provides another endpoint if you want to create or update a Test Executions and have control over all Test Execution fields. It allows you to send two JSON files, the normal Xray JSON result and a JSON similar to the one Jira uses to create/update issues. For more information about that second format, check the Jira documentation here. Note that in this endpoint the `info` property in the Xray Json result part will be ignored.

Import the execution results present in query variable "***executionResults***".

Request

Example 1: new Test Execution

Example Input

```
{
    "info" : {
        "summary" : "Execution of automated tests for release v1.3",
                "description" : "This execution is automatically created when importing execution results
from an external source",
                "version" : "v1.3",
                "user" : "admin",
                "revision" : "1.0.42134",
                "startDate" : "2014-08-30T11:47:35+01:00",
                "finishDate" : "2014-08-30T11:53:00+01:00",
                "testPlanKey" : "DEMO-100",
                "testEnvironments": ["iOS", "Android"]
    },
    "tests" : [
        {
            "testKey" : "DEMO-6",
            "start" : "2014-08-30T11:47:35+01:00",
            "finish" : "2014-08-30T11:50:56+01:00",
            "comment" : "Successful execution",
            "status" : "PASSED"
        },
        {
            "testKey" : "DEMO-7",
            "start" : "2014-08-30T11:51:00+01:00",
            "finish" : "2014-08-30T11:52:30+01:00",
            "comment" : "Execution failed. Example #5 FAIL.",
            "status" : "FAILED",
                        "evidences" : [
                    {
                        "data":
"iVBORw0KGgoAAAANSUhEUgAABkIAAAO9CAYAAADezXv6AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEn(...base64
file enconding)",
                                        "filename": "image21.jpg",
                    "contentType": "image/jpeg"
                }
                        ],
            "examples" : [
                "PASSED",
                "PASSED",
                "PASSED",
                "PASSED",
                "FAILED"
            ],
                        "steps": [
                {
                    "status": "PASSED",
                                        "actualResult" : "Step Passed with Status Code 200",
                    "comment": "Comment on Test Step Result 1",
                    "evidences" : [
                        {
                            "data":
"iVBORw0KGgoAAAANSUhEUgAABkIAAAO9CAYAAADezXv6AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEn(...base64
file enconding)",
                                                                "filename": "image22.jpg",
                        "contentType": "image/jpeg"
                        }
                                        ]
                                }
                        ],
                    "defects" : [
                                        "DEMO-10",
                                        "DEMO-11"
                ]
        }
    ]
}
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Example 2: update existing Test Execution

Example Input

```
{
    "testExecutionKey": "DEMO-1206",
    "info" : {
        "summary" : "Execution of automated tests for release v1.3",
                "description" : "This execution is automatically created when importing execution results
from an external source",
                "version" : "v1.3",
                "user" : "admin",
                "revision" : "1.0.42134",
                "startDate" : "2014-08-30T11:47:35+01:00",
                "finishDate" : "2014-08-30T11:53:00+01:00",
                "testPlanKey" : "DEMO-100",
                "testEnvironments": ["iOS", "Android"]
    },
    "tests" : [
        {
            "testKey" : "DEMO-6",
            "start" : "2014-08-30T11:47:35+01:00",
            "finish" : "2014-08-30T11:50:56+01:00",
            "comment" : "Successful execution",
            "status" : "PASSED"
        }
     ]
}
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/multipart

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** No execution results were provided**.**

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

Cucumber JSON results

After executing Cucumber features, you must import the outputted JSON execution results to Jira using the following endpoint:

Import the execution results created with the Cucumber JSON output formatter. For more information please check the Cucumber reports documentation (example here).

Request

Example

Example Input


```
[
  {
    "keyword": "Feature",
    "name": "Arithmetic Operations",
    "line": 3,
    "description": "",
    "tags": [
      {
        "name": "@DEMO-48",
        "line": 1
      },
      {
        "name": "@REQ_DEMO-45",
        "line": 2
      }
    ],
    "id": "arithmetic-operations",
    "uri": "features/1_DEMO-45.feature",
    "elements": [
      {
        "comments": [
          {
            "value": "#In order to avoid silly mistakes",
            "line": 4
          },
          {
            "value": "#As a math idiot ",
            "line": 5
          },
          {
            "value": "#I want to be told the result of basic arithmetic operations between two numbers",
            "line": 6
          }
        ],
        "keyword": "Scenario Outline",
        "name": "Add two Numbers",
        "line": 18,
        "description": "",
        "tags": [
          {
            "name": "@TEST_DEMO-47",
            "line": 9
          }
        ],
        "id": "arithmetic-operations;add-two-numbers;;2",
        "type": "scenario",
        "steps": [
          {
                        "embeddings": [
              {
                                "mime_type": "text/plain",
                                "data": "{data base64}"
              },{
                                "mime_type": "text/plain",
                                "data": "{data base64}"
              }],
            "keyword": "Given ",
            "name": "I have entered 20 into the calculator",
            "line": 11,
            "match": {
              "arguments": [
                {
                  "offset": 15,
                  "val": "20"
                }
              ],
              "location": "features/step_definitions/calculator_steps.rb:14"
            },
```

```
        "result": {
          "status": "passed",
          "duration": 487000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 30 into the calculator",
        "line": 12,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "30"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 340000
        }
      },
      {
        "keyword": "When ",
        "name": "I press add",
        "line": 13,
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "add"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 327000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 50 on the screen",
        "line": 14,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "50"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 11723000
        }
      }
    ]
  },
  {
    "comments": [
      {
        "value": "#In order to avoid silly mistakes",
        "line": 4
      },
      {
        "value": "#As a math idiot ",
        "line": 5
      },
```

```
    {
      "value": "#I want to be told the result of basic arithmetic operations between two numbers",
      "line": 6
    }
  ],
  "keyword": "Scenario Outline",
  "name": "Add two Numbers",
  "line": 19,
  "description": "",
  "tags": [
    {
      "name": "@TEST_DEMO-47",
      "line": 9
    }
  ],
  "id": "arithmetic-operations;add-two-numbers;;3",
  "type": "scenario",
  "steps": [
    {
      "keyword": "Given ",
      "name": "I have entered 2 into the calculator",
      "line": 11,
      "match": {
        "arguments": [
          {
            "offset": 15,
            "val": "2"
          }
        ],
        "location": "features/step_definitions/calculator_steps.rb:14"
      },
      "result": {
        "status": "passed",
        "duration": 992000
      }
    },
    {
      "keyword": "And ",
      "name": "I have entered 5 into the calculator",
      "line": 12,
      "match": {
        "arguments": [
          {
            "offset": 15,
            "val": "5"
          }
        ],
        "location": "features/step_definitions/calculator_steps.rb:14"
      },
      "result": {
        "status": "passed",
        "duration": 775000
      }
    },
    {
      "keyword": "When ",
      "name": "I press add",
      "line": 13,
      "match": {
        "arguments": [
          {
            "offset": 8,
            "val": "add"
          }
        ],
        "location": "features/step_definitions/calculator_steps.rb:18"
      },
      "result": {
        "status": "passed",
        "duration": 322000
      }
```

```
      },
      {
        "keyword": "Then ",
        "name": "the result should be 7 on the screen",
        "line": 14,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "7"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 423000
        }
      }
    ]
  },
  {
    "comments": [
      {
        "value": "#In order to avoid silly mistakes",
        "line": 4
      },
      {
        "value": "#As a math idiot ",
        "line": 5
      },
      {
        "value": "#I want to be told the result of basic arithmetic operations between two numbers",
        "line": 6
      }
    ],
    "keyword": "Scenario Outline",
    "name": "Add two Numbers",
    "line": 20,
    "description": "",
    "tags": [
      {
        "name": "@TEST_DEMO-47",
        "line": 9
      }
    ],
    "id": "arithmetic-operations;add-two-numbers;;4",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 0 into the calculator",
        "line": 11,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "0"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 384000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 40 into the calculator",
        "line": 12,
```

```
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "40"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 313000
        }
      },
      {
        "keyword": "When ",
        "name": "I press add",
        "line": 13,
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "add"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 280000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 40 on the screen",
        "line": 14,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "40"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 350000
        }
      }
    ]
  },
  {
    "keyword": "Scenario Outline",
    "name": "Divide Two Numbers",
    "line": 32,
    "description": "",
    "tags": [
      {
        "name": "@TEST_DEMO-46",
        "line": 23
      }
    ],
    "id": "arithmetic-operations;divide-two-numbers;;2",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 8 into the calculator",
        "line": 25,
        "match": {
```

```
          "arguments": [
            {
              "offset": 15,
              "val": "8"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 344000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 4 into the calculator",
        "line": 26,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "4"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 292000
        }
      },
      {
        "keyword": "When ",
        "name": "I press divide",
        "line": 27,
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "divide"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 291000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 2 on the screen",
        "line": 28,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "2"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 320000
        }
      }
    ]
  },
  {
    "keyword": "Scenario Outline",
```

```
"name": "Divide Two Numbers",
"line": 33,
"description": "",
"tags": [
  {
    "name": "@TEST_DEMO-46",
    "line": 23
  }
],
"id": "arithmetic-operations;divide-two-numbers;;3",
"type": "scenario",
"steps": [
  {
    "keyword": "Given ",
    "name": "I have entered 12 into the calculator",
    "line": 25,
    "match": {
      "arguments": [
        {
          "offset": 15,
          "val": "12"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:14"
    },
    "result": {
      "status": "passed",
      "duration": 1102000
    }
  },
  {
    "keyword": "And ",
    "name": "I have entered 3 into the calculator",
    "line": 26,
    "match": {
      "arguments": [
        {
          "offset": 15,
          "val": "3"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:14"
    },
    "result": {
      "status": "passed",
      "duration": 891000
    }
  },
  {
    "keyword": "When ",
    "name": "I press divide",
    "line": 27,
    "match": {
      "arguments": [
        {
          "offset": 8,
          "val": "divide"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:18"
    },
    "result": {
      "status": "passed",
      "duration": 291000
    }
  },
  {
    "keyword": "Then ",
    "name": "the result should be 4 on the screen",
    "line": 28,
    "match": {
```

```
        "arguments": [
          {
            "offset": 21,
            "val": "4"
          }
        ],
        "location": "features/step_definitions/calculator_steps.rb:22"
      },
      "result": {
        "status": "passed",
        "duration": 339000
      }
    }
  ]
},
{
  "keyword": "Scenario Outline",
  "name": "Divide Two Numbers",
  "line": 34,
  "description": "",
  "tags": [
    {
      "name": "@TEST_DEMO-46",
      "line": 23
    }
  ],
  "id": "arithmetic-operations;divide-two-numbers;;4",
  "type": "scenario",
  "steps": [
    {
      "keyword": "Given ",
      "name": "I have entered 3 into the calculator",
      "line": 25,
      "match": {
        "arguments": [
          {
            "offset": 15,
            "val": "3"
          }
        ],
        "location": "features/step_definitions/calculator_steps.rb:14"
      },
      "result": {
        "status": "passed",
        "duration": 304000
      }
    },
    {
      "keyword": "And ",
      "name": "I have entered 1 into the calculator",
      "line": 26,
      "match": {
        "arguments": [
          {
            "offset": 15,
            "val": "1"
          }
        ],
        "location": "features/step_definitions/calculator_steps.rb:14"
      },
      "result": {
        "status": "passed",
        "duration": 309000
      }
    },
    {
      "keyword": "When ",
      "name": "I press divide",
      "line": 27,
      "match": {
        "arguments": [
```

```
                {
                  "offset": 8,
                  "val": "divide"
                }
              ],
              "location": "features/step_definitions/calculator_steps.rb:18"
            },
            "result": {
              "status": "passed",
              "duration": 257000
            }
          },
          {
            "keyword": "Then ",
            "name": "the result should be 5 on the screen",
            "line": 28,
            "match": {
              "arguments": [
                {
                  "offset": 21,
                  "val": "5"
                }
              ],
              "location": "features/step_definitions/calculator_steps.rb:22"
            },
            "result": {
              "status": "passed",
              "duration": 840000
            }
          }
        ]
      }
    ]
  }
]
```


Example Request

curl -H "Content-Type: application/json" -X POST -H "Authorization: Bearer $token" --data @"data.json" https://xray.cloud.getxray.app/api/v2/import/execution/cucumber

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output


```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```


400 BAD_REQUEST : **application/json :** No execution results were provided**.**

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

Cucumber JSON results Multipart

Xray provides another endpoint if you want to create new Test Executions and have control over newly-created Test Execution fields. It allows you to send two JSON files, the normal Cucumber result JSON and a JSON similar to the one Jira uses to create new issues. For more information about that second format, check the Jira documentation here.

Import the execution results created with the Cucumber JSON output formatter. For more information please check the Cucumber reports documentation (example here).

Request

Example

Example Input


```
[
  {
    "keyword": "Feature",
    "name": "Arithmetic Operations",
    "line": 3,
    "description": "",
    "tags": [
      {
        "name": "@DEMO-48",
        "line": 1
      },
      {
        "name": "@REQ_DEMO-45",
        "line": 2
      }
    ],
    "id": "arithmetic-operations",
    "uri": "features/1_DEMO-45.feature",
    "elements": [
      {
        "comments": [
          {
            "value": "#In order to avoid silly mistakes",
            "line": 4
          },
          {
            "value": "#As a math idiot ",
            "line": 5
          },
          {
            "value": "#I want to be told the result of basic arithmetic operations between two numbers",
            "line": 6
          }
        ],
        "keyword": "Scenario Outline",
        "name": "Add two Numbers",
        "line": 18,
        "description": "",
        "tags": [
          {
            "name": "@TEST_DEMO-47",
            "line": 9
          }
        ],
        "id": "arithmetic-operations;add-two-numbers;;2",
        "type": "scenario",
        "steps": [
          {
                      "embeddings": [
              {
                                "mime_type": "text/plain",
                                "data": "{data base64}"
              },{
                                "mime_type": "text/plain",
                                "data": "{data base64}"
```

```
      }],
    "keyword": "Given ",
    "name": "I have entered 20 into the calculator",
    "line": 11,
    "match": {
      "arguments": [
        {
          "offset": 15,
          "val": "20"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:14"
    },
    "result": {
      "status": "passed",
      "duration": 487000
    }
  },
  {
    "keyword": "And ",
    "name": "I have entered 30 into the calculator",
    "line": 12,
    "match": {
      "arguments": [
        {
          "offset": 15,
          "val": "30"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:14"
    },
    "result": {
      "status": "passed",
      "duration": 340000
    }
  },
  {
    "keyword": "When ",
    "name": "I press add",
    "line": 13,
    "match": {
      "arguments": [
        {
          "offset": 8,
          "val": "add"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:18"
    },
    "result": {
      "status": "passed",
      "duration": 327000
    }
  },
  {
    "keyword": "Then ",
    "name": "the result should be 50 on the screen",
    "line": 14,
    "match": {
      "arguments": [
        {
          "offset": 21,
          "val": "50"
        }
      ],
      "location": "features/step_definitions/calculator_steps.rb:22"
    },
    "result": {
      "status": "passed",
      "duration": 11723000
    }
```

```
      }
    ]
  },
  {
    "comments": [
      {
        "value": "#In order to avoid silly mistakes",
        "line": 4
      },
      {
        "value": "#As a math idiot ",
        "line": 5
      },
      {
        "value": "#I want to be told the result of basic arithmetic operations between two numbers",
        "line": 6
      }
    ],
    "keyword": "Scenario Outline",
    "name": "Add two Numbers",
    "line": 19,
    "description": "",
    "tags": [
      {
        "name": "@TEST_DEMO-47",
        "line": 9
      }
    ],
    "id": "arithmetic-operations;add-two-numbers;;3",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 2 into the calculator",
        "line": 11,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "2"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 992000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 5 into the calculator",
        "line": 12,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "5"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 775000
        }
      },
      {
        "keyword": "When ",
        "name": "I press add",
        "line": 13,
```

```
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "add"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 322000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 7 on the screen",
        "line": 14,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "7"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 423000
        }
      }
    ]
  },
  {
    "comments": [
      {
        "value": "#In order to avoid silly mistakes",
        "line": 4
      },
      {
        "value": "#As a math idiot ",
        "line": 5
      },
      {
        "value": "#I want to be told the result of basic arithmetic operations between two numbers",
        "line": 6
      }
    ],
    "keyword": "Scenario Outline",
    "name": "Add two Numbers",
    "line": 20,
    "description": "",
    "tags": [
      {
        "name": "@TEST_DEMO-47",
        "line": 9
      }
    ],
    "id": "arithmetic-operations;add-two-numbers;;4",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 0 into the calculator",
        "line": 11,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "0"
```

```
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 384000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 40 into the calculator",
        "line": 12,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "40"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 313000
        }
      },
      {
        "keyword": "When ",
        "name": "I press add",
        "line": 13,
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "add"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 280000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 40 on the screen",
        "line": 14,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "40"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 350000
        }
      }
    ]
  },
  {
    "keyword": "Scenario Outline",
    "name": "Divide Two Numbers",
    "line": 32,
    "description": "",
    "tags": [
```

```
      {
        "name": "@TEST_DEMO-46",
        "line": 23
      }
    ],
    "id": "arithmetic-operations;divide-two-numbers;;2",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 8 into the calculator",
        "line": 25,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "8"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 344000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 4 into the calculator",
        "line": 26,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "4"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 292000
        }
      },
      {
        "keyword": "When ",
        "name": "I press divide",
        "line": 27,
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "divide"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 291000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 2 on the screen",
        "line": 28,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "2"
```

```
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 320000
        }
      }
    ]
  },
  {
    "keyword": "Scenario Outline",
    "name": "Divide Two Numbers",
    "line": 33,
    "description": "",
    "tags": [
      {
        "name": "@TEST_DEMO-46",
        "line": 23
      }
    ],
    "id": "arithmetic-operations;divide-two-numbers;;3",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 12 into the calculator",
        "line": 25,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "12"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 1102000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 3 into the calculator",
        "line": 26,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "3"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 891000
        }
      },
      {
        "keyword": "When ",
        "name": "I press divide",
        "line": 27,
        "match": {
          "arguments": [
            {
              "offset": 8,
              "val": "divide"
            }
```

```
          ],
          "location": "features/step_definitions/calculator_steps.rb:18"
        },
        "result": {
          "status": "passed",
          "duration": 291000
        }
      },
      {
        "keyword": "Then ",
        "name": "the result should be 4 on the screen",
        "line": 28,
        "match": {
          "arguments": [
            {
              "offset": 21,
              "val": "4"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:22"
        },
        "result": {
          "status": "passed",
          "duration": 339000
        }
      }
    ]
  },
  {
    "keyword": "Scenario Outline",
    "name": "Divide Two Numbers",
    "line": 34,
    "description": "",
    "tags": [
      {
        "name": "@TEST_DEMO-46",
        "line": 23
      }
    ],
    "id": "arithmetic-operations;divide-two-numbers;;4",
    "type": "scenario",
    "steps": [
      {
        "keyword": "Given ",
        "name": "I have entered 3 into the calculator",
        "line": 25,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "3"
            }
          ],
          "location": "features/step_definitions/calculator_steps.rb:14"
        },
        "result": {
          "status": "passed",
          "duration": 304000
        }
      },
      {
        "keyword": "And ",
        "name": "I have entered 1 into the calculator",
        "line": 26,
        "match": {
          "arguments": [
            {
              "offset": 15,
              "val": "1"
            }
          ],
```

```
            "location": "features/step_definitions/calculator_steps.rb:14"
          },
          "result": {
            "status": "passed",
            "duration": 309000
          }
        },
        {
          "keyword": "When ",
          "name": "I press divide",
          "line": 27,
          "match": {
            "arguments": [
              {
                "offset": 8,
                "val": "divide"
              }
            ],
            "location": "features/step_definitions/calculator_steps.rb:18"
          },
          "result": {
            "status": "passed",
            "duration": 257000
          }
        },
        {
          "keyword": "Then ",
          "name": "the result should be 5 on the screen",
          "line": 28,
          "match": {
            "arguments": [
              {
                "offset": 21,
                "val": "5"
              }
            ],
            "location": "features/step_definitions/calculator_steps.rb:22"
          },
          "result": {
            "status": "passed",
            "duration": 840000
          }
        }
      ]
    }
  ]
}
]
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/cucumber/multipart

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** No execution results were provided**.**

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

JUnit XML results

After executing JUnit tests, you must import the outputted XML execution results to Jira using the following endpoint:

Import the execution results created with the JUnit XML output formatter. For more information, please check the documentation about JUnit integration.

Request

PATH PARAMETERS

parameter	type	description
`projectKey`	*String*	- key of the project where the test execution (if the **testExecKey** parameter wasn't provided) and the tests (if they aren't created yet) are going to be created.
testExecKey	String	- key of the Test Execution.
testPlanKey	String	- key of the Test Plan; if you specify the Test Plan, the Tests will be added automatically to the Test Plan if they're not part of it.
testEnvironments	String	- a string containing a list of test environments separated by ";"
revision	String	- source code and documentation version used in the test execution.
fixVersion	String	- the Fix Version associated with the test execution (it supports only one value).

Example

JUnit Report XML

```
<?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>
```

Example Request

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/junit?projectKey=XTP

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/junit?testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/junit?projectKey=XTP&testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/junit?projectKey=XTP&testPlanKey=XTP-12&revision=v2.1.0

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** Returns the error.

401 UNAUTHORIZED : **application/json** : The token is invalid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

JUnit XML results Multipart

Xray provides another endpoint if you want to have control over the fields of newly created Test Executions and Tests. The endpoint accepts one XML file (the JUnit report) and two JSON files similar to the one Jira uses to create new issues, one with the fields of the Test Execution and the other with the fields of the Test. For more information about that JSON format, check the Jira documentation here. Xray will ignore the issue type and the summary fields on the Test JSON file since these are provided by the importation.

Import the execution results created with the JUnit XML output formatter. For more information, please check the documentation about JUnit integration.

Request

Example

JUnit Report XML

```
<?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>
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Info JSON (Test)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "labels" : ["Server","JUnit"]
    }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.xml -F testInfo=@testIssueFields.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/junit/multipart

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** Returns the error.

401 UNAUTHORIZED : **application/json** : The token is invalid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

NUnit XML results

After executing NUnit tests, you must import the outputted XML execution results to Jira using the following endpoint:

Import the execution results created with the NUnit XML output formatter. For more information please check the documentation about NUnit integration.

Request

PATH PARAMETERS

parameter	type	description
`projectKey`	*String*	- key of the project where the Test Execution (if the **testExecKey** parameter wasn't provided) and the tests (if they aren't created yet) are going to be created.
testExecKey	String	- key of the Test Execution.
testPlanKey	String	- key of the Test Plan; if you specify the Test Plan, the Tests will be added automatically to the Test Plan if they're not part of it.
testEnvironments	String	- a string containing a list of test environments separated by ";"
revision	String	- source code and documentation version used in the test execution.
fixVersion	String	- the Fix Version associated with the test execution (it supports only one value).

Example

NUnit Report XML

```
<?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>
```

Example Request

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/nunit?projectKey=XTP

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/nunit?testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/nunit?projectKey=XTP&testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/nunit?projectKey=XTP&testPlanKey=XTP-12&revision=123&fixVersion=v2.1.0

Responses

200 OK : ***application/json*** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : **application/json** : *The token is invalid.*

500 INTERNAL SERVER ERROR : **application/json** : *An internal error occurred when importing execution results.*

NUnit XML results Multipart

Xray provides another endpoint if you want to have control over the fields of newly created Test Executions and Tests. The endpoint accepts one XML file (the NUnit report) and two JSON files similar to the one Jira uses to create new issues, one with the fields of the Test Execution and the other with the fields of the Test. For more information about that JSON format, check the Jira documentation here. Xray will ignore the issue type and the summary fields on the Test JSON file since these are provided by the importation.

Import the execution results created with the NUnit XML output formatter. For more information please check the documentation about NUnit integration.

Request

Example

NUnit Report XML

```
<?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>
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Info JSON (Test)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "labels" : ["Server","NUnit"]
    }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.xml -F testInfo=@testIssueFields.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/nunit/multipart

Responses

200 OK : ***application/json*** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : ***application/json*** *: The token is invalid.*

500 INTERNAL SERVER ERROR : ***application/json*** *: An internal error occurred when importing execution results.*

xUnit XML results

After executing xUnit tests, you must import the outputted XML execution results to Jira using the following endpoint:

Import the execution results created with the xUnit XML output formatter. For more information please check the documentation about xUnit integration.

Request

PATH PARAMETERS

parameter	type	description
`projectKey`	*String*	- key of the project where the Test Execution (if the **testExecKey** parameter wasn't provided) and the tests (if they aren't created yet) are going to be created.
testExecKey	String	- key of the Test Execution.
testPlanKey	String	- key of the Test Plan; if you specify the Test Plan, the Tests will be added automatically to the Test Plan if they're not part of it.
testEnvironments	String	- a string containing a list of test environments separated by ";"
revision	String	- source code and documentation version used in the test execution.
fixVersion	String	- the Fix Version associated with the test execution (it supports only one value).

Example

xUnit Report XML

```
<?xml version="1.0" encoding="utf-8"?>
<assemblies timestamp="07/31/2018 14:58:48">
    <assembly name="C:\Users\smsf\Documents\Visual Studio
2015\Projects\xUnitDemo\xUnitDemo\bin\Debug\xUnitDemo.DLL" environment="64-bit .NET 4.0.30319.42000
[collection-per-class, parallel (1 threads)]" test-framework="xUnit.net 2.3.1.3858" run-date="2018-07-31" run-
time="14:58:47" config-file="C:\Users\smsf\Documents\Visual Studio 2015\Projects\xUnitDemo\packages\xunit.
runner.console.2.4.0\tools\net452\xunit.console.exe.Config" total="15" passed="14" failed="1" skipped="0"
time="0.257" errors="0">
        <errors />
        <collection total="2" passed="1" failed="1" skipped="0" name="Test collection for xUnitDemo.
SimpleTests" time="0.070">
            <test name="xUnitDemo.SimpleTests.PassingTest" type="xUnitDemo.SimpleTests" method="PassingTest"
time="0.0636741" result="Pass">
                <traits>
                    <trait name="test" value="CALC-1" />
                    <trait name="requirement" value="CALC-1" />
                    <trait name="labels" value="core UI" />
                </traits>
            </test>
            <test name="xUnitDemo.SimpleTests.FailingTest" type="xUnitDemo.SimpleTests" method="FailingTest"
time="0.0059474" result="Fail">
                <failure exception-type="Xunit.Sdk.EqualException">
                    <message><![CDATA[Assert.Equal() Failure\r\nExpected: 5\r\nActual: 4]]></message>
                    <stack-trace><![CDATA[ at xUnitDemo.SimpleTests.FailingTest() in C:
\Users\smsf\documents\visual studio 2015\Projects\xUnitDemo\xUnitDemo\SimpleTests.cs:line 30]]></stack-trace>
                </failure>
            </test>
        </collection>
        <collection total="13" passed="13" failed="0" skipped="0" name="Test collection for xUnitDemo.
CalculatorTests" time="0.001">
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: 1, value2: 2, expected: 3)"
type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0002722" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: -4, value2: -6, expected:
-10)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0000248" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: -2, value2: 2, expected: 0)"
type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0000028" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: -2147483648, value2: -1,
expected: 2147483647)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0000017"
result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryMemberDataMethod(value1: 1, value2: 2,
expected: 3)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryMemberDataMethod" time="0.0001179" result="
Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryMemberDataMethod(value1: -4, value2: -6,
expected: -10)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryMemberDataMethod" time="0.0000054"
result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryMemberDataMethod(value1: -2, value2: 2,
expected: 0)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryMemberDataMethod" time="0.0000024" result="
Pass" />
            <test name="xUnitDemo.CalculatorTests.PassingTest" type="xUnitDemo.CalculatorTests" method="
PassingTest" time="0.0000952" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: 1, value2: 2, expected: 3)" type="
xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.0001095" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: -4, value2: -6, expected: -10)" type="
xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.0000083" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: -2, value2: 2, expected: 0)" type="
xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.000002" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: -2147483648, value2: -1, expected:
2147483647)" type="xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.0000017" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.SkippedTest" type="xUnitDemo.CalculatorTests" method="
SkippedTest" time="" result="Skip">
                <reason><![CDATA[this test was skipped]]></reason>
            </test>
        </collection>
    </assembly>
</assemblies>
```

Example Request

```
curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token"  --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/xunit?projectKey=XTP

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token"  --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/xunit?testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token"  --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/xunit?projectKey=XTP&testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token"  --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/xunit?projectKey=XTP&testPlanKey=XTP-12&revision=123&fixVersion=v2.1.0
```

Responses

200 OK : ***application/json*** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : ***application/json*** *: The token is invalid.*

500 INTERNAL SERVER ERROR : ***application/json*** *: An internal error occurred when importing execution results.*

xUnit XML results Multipart

Xray provides another endpoint if you want to have control over the fields of newly created Test Executions and Tests. The endpoint accepts one XML file (the xUnit report) and two JSON files similar to the one Jira uses to create new issues, one with the fields of the Test Execution and the other with the fields of the Test. For more information about that JSON format, check the Jira documentation here. Xray will ignore the issue type and the summary fields on the Test JSON file since these are provided by the importation.

Import the execution results created with the xUnit XML output formatter. For more information please check the documentation about xUnit integration.

Request

Example

xUnit Report XML

```
<?xml version="1.0" encoding="utf-8"?>
<assemblies timestamp="07/31/2018 14:58:48">
    <assembly name="C:\Users\smsf\Documents\Visual Studio
2015\Projects\xUnitDemo\xUnitDemo\bin\Debug\xUnitDemo.DLL" environment="64-bit .NET 4.0.30319.42000
[collection-per-class, parallel (1 threads)]" test-framework="xUnit.net 2.3.1.3858" run-date="2018-07-31" run-
time="14:58:47" config-file="C:\Users\smsf\Documents\Visual Studio 2015\Projects\xUnitDemo\packages\xunit.
runner.console.2.4.0\tools\net452\xunit.console.exe.Config" total="15" passed="14" failed="1" skipped="0"
time="0.257" errors="0">
        <errors />
        <collection total="2" passed="1" failed="1" skipped="0" name="Test collection for xUnitDemo.
SimpleTests" time="0.070">
            <test name="xUnitDemo.SimpleTests.PassingTest" type="xUnitDemo.SimpleTests" method="PassingTest"
time="0.0636741" result="Pass">
                <traits>
                    <trait name="test" value="CALC-1" />
                    <trait name="requirement" value="CALC-1" />
                    <trait name="labels" value="core UI" />
                </traits>
            </test>
            <test name="xUnitDemo.SimpleTests.FailingTest" type="xUnitDemo.SimpleTests" method="FailingTest"
time="0.0059474" result="Fail">
                <failure exception-type="Xunit.Sdk.EqualException">
                    <message><![CDATA[Assert.Equal() Failure\r\nExpected: 5\r\nActual: 4]]></message>
                    <stack-trace><![CDATA[ at xUnitDemo.SimpleTests.FailingTest() in C:
\Users\smsf\documents\visual studio 2015\Projects\xUnitDemo\xUnitDemo\SimpleTests.cs:line 30]]></stack-trace>
                </failure>
            </test>
        </collection>
        <collection total="13" passed="13" failed="0" skipped="0" name="Test collection for xUnitDemo.
CalculatorTests" time="0.001">
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: 1, value2: 2, expected: 3)"
type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0002722" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: -4, value2: -6, expected:
-10)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0000248" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: -2, value2: 2, expected: 0)"
type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0000028" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryClassData(value1: -2147483648, value2: -1,
expected: 2147483647)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryClassData" time="0.0000017"
result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryMemberDataMethod(value1: 1, value2: 2,
expected: 3)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryMemberDataMethod" time="0.0001179" result="
Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryMemberDataMethod(value1: -4, value2: -6,
expected: -10)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryMemberDataMethod" time="0.0000054"
result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheoryMemberDataMethod(value1: -2, value2: 2,
expected: 0)" type="xUnitDemo.CalculatorTests" method="CanAddTheoryMemberDataMethod" time="0.0000024" result="
Pass" />
            <test name="xUnitDemo.CalculatorTests.PassingTest" type="xUnitDemo.CalculatorTests" method="
PassingTest" time="0.0000952" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: 1, value2: 2, expected: 3)" type="
xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.0001095" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: -4, value2: -6, expected: -10)" type="
xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.0000083" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: -2, value2: 2, expected: 0)" type="
xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.000002" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.CanAddTheory(value1: -2147483648, value2: -1, expected:
2147483647)" type="xUnitDemo.CalculatorTests" method="CanAddTheory" time="0.0000017" result="Pass" />
            <test name="xUnitDemo.CalculatorTests.SkippedTest" type="xUnitDemo.CalculatorTests" method="
SkippedTest" time="" result="Skip">
                <reason><![CDATA[this test was skipped]]></reason>
            </test>
        </collection>
    </assembly>
</assemblies>
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Info JSON (Test)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "labels" : ["Server","xUnit"]
    }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.xml -F testInfo=@testIssueFields.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/xunit/multipart

Responses

200 OK : ***application/json*** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : ***application/json*** *: The token is invalid.*

500 INTERNAL SERVER ERROR : ***application/json*** *: An internal error occurred when importing execution results.*

TestNG XML results

After executing TestNG tests, you must import the outputted XML execution results to Jira using the following endpoint:

Import the execution results created with the TestNG XML output formatter. For more information please check the documentation about TestNG integration.

Request

QUERY PARAMETERS

parameter	type	description
`projectKey`	*String*	- key of the project where the Test Execution (if the **testExecKey** parameter wasn't provided) and the tests (if they aren't created yet) are going to be created.
testExecKey	String	- key of the Test Execution.
testPlanKey	String	- key of the Test Plan; if you specify the Test Plan, the Tests will be added automatically to the Test Plan if they're not part of it.
testEnvironments	String	- a string containing a list of test environments separated by ";"
revision	String	- source code and documentation version used in the test execution.
fixVersion	String	- the Fix Version associated with the test execution (it supports only one value).

Example

TestNG XML Report

```
<?xml version="1.0" encoding="UTF-8"?>
<testng-results skipped="0" failed="2" ignored="0" total="8" passed="6">
  <reporter-output>
  </reporter-output>
  <suite name="TestAll" duration-ms="33" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
    <groups>
    </groups>
    <test name="calculator" duration-ms="33" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:
00Z">
      <class name="com.xpand.java.CalcTest">
        <test-method status="PASS" signature="setUp()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]" name="
setUp" is-config="true" duration-ms="9" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
        </test-method> <!-- setUp -->
        <test-method status="PASS" signature="CanAddNumbers()[pri:0, instance:com.xpand.java.
CalcTest@36d4b5c]" name="CanAddNumbers" duration-ms="2" started-at="2018-03-06T11:53:00Z" finished-at="2018-
03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbers -->
        <test-method status="PASS" signature="CanAddNumbersFromGivenData(int, int, int)[pri:0, instance:com.
xpand.java.CalcTest@36d4b5c]" name="CanAddNumbersFromGivenData" duration-ms="0" started-at="2018-03-06T11:53:
00Z" data-provider="ValidDataProvider" finished-at="2018-03-06T11:53:00Z">
          <params>
            <param index="0">
              <value>
```

```
              <![CDATA[1]]>
            </value>
          </param>
          <param index="1">
            <value>
              <![CDATA[2]]>
            </value>
          </param>
          <param index="2">
            <value>
              <![CDATA[3]]>
            </value>
          </param>
        </params>
        <reporter-output>
        </reporter-output>
        <attributes>
          <attribute name="test">
            <![CDATA[]]>
          </attribute> <!-- test -->
          <attribute name="requirement">
            <![CDATA[CALC-1235]]>
          </attribute> <!-- requirement -->
          <attribute name="labels">
            <![CDATA[core]]>
          </attribute> <!-- labels -->
        </attributes>
      </test-method> <!-- CanAddNumbersFromGivenData -->
      <test-method status="FAIL" signature="CanAddNumbersFromGivenData(int, int, int)[pri:0, instance:com.
xpand.java.CalcTest@36d4b5c]" name="CanAddNumbersFromGivenData" duration-ms="1" started-at="2018-03-06T11:53:
00Z" data-provider="ValidDataProvider" finished-at="2018-03-06T11:53:00Z">
        <params>
          <param index="0">
            <value>
              <![CDATA[2]]>
            </value>
          </param>
          <param index="1">
            <value>
              <![CDATA[3]]>
            </value>
          </param>
          <param index="2">
            <value>
              <![CDATA[4]]>
            </value>
          </param>
        </params>
        <exception class="java.lang.AssertionError">
          <message>
            <![CDATA[expected [4] but found [5]]]>
          </message>
          <full-stacktrace>
            <![CDATA[java.lang.AssertionError: expected [4] but found [5]
      at org.testng.Assert.fail(Assert.java:93)
      at org.testng.Assert.failNotEquals(Assert.java:512)
      at org.testng.Assert.assertEqualsImpl(Assert.java:134)
      at org.testng.Assert.assertEquals(Assert.java:115)
      at org.testng.Assert.assertEquals(Assert.java:388)
      at org.testng.Assert.assertEquals(Assert.java:398)
      at com.xpand.java.CalcTest.CanAddNumbersFromGivenData(CalcTest.java:40)
      at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
      at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
      at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
      at java.lang.reflect.Method.invoke(Method.java:498)
      at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:108)
      at org.testng.internal.Invoker.invokeMethod(Invoker.java:661)
      at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:869)
      at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:1193)
      at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:126)
      at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:109)
```

```
        at org.testng.TestRunner.privateRun(TestRunner.java:744)
        at org.testng.TestRunner.run(TestRunner.java:602)
        at org.testng.SuiteRunner.runTest(SuiteRunner.java:380)
        at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:375)
        at org.testng.SuiteRunner.privateRun(SuiteRunner.java:340)
        at org.testng.SuiteRunner.run(SuiteRunner.java:289)
        at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
        at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:86)
        at org.testng.TestNG.runSuitesSequentially(TestNG.java:1301)
        at org.testng.TestNG.runSuitesLocally(TestNG.java:1226)
        at org.testng.TestNG.runSuites(TestNG.java:1144)
        at org.testng.TestNG.run(TestNG.java:1115)
        at org.apache.maven.surefire.testng.TestNGExecutor.run(TestNGExecutor.java:283)
        at org.apache.maven.surefire.testng.TestNGXmlTestSuite.execute(TestNGXmlTestSuite.java:75)
        at org.apache.maven.surefire.testng.TestNGProvider.invoke(TestNGProvider.java:120)
        at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:
373)
        at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:334)
        at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:119)
        at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:407)
]]>
            </full-stacktrace>
          </exception> <!-- java.lang.AssertionError -->
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbersFromGivenData -->
        <test-method status="PASS" signature="CanAddNumbersFromGivenData(int, int, int)[pri:0, instance:com.
xpand.java.CalcTest@36d4b5c]" name="CanAddNumbersFromGivenData" duration-ms="0" started-at="2018-03-06T11:53:
00Z" data-provider="ValidDataProvider" finished-at="2018-03-06T11:53:00Z">
          <params>
            <param index="0">
              <value>
                <![CDATA[-1]]>
              </value>
            </param>
            <param index="1">
              <value>
                <![CDATA[1]]>
              </value>
            </param>
            <param index="2">
              <value>
                <![CDATA[0]]>
              </value>
            </param>
          </params>
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbersFromGivenData -->
```

```
        <test-method status="FAIL" signature="CanDoStuff()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="CanDoStuff" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <exception class="java.lang.AssertionError">
            <message>
              <![CDATA[null]]>
            </message>
            <full-stacktrace>
              <![CDATA[java.lang.AssertionError: null
        at org.testng.Assert.fail(Assert.java:93)
        at org.testng.Assert.assertNotEquals(Assert.java:897)
        at org.testng.Assert.assertNotEquals(Assert.java:902)
        at com.xpand.java.CalcTest.CanDoStuff(CalcTest.java:86)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:108)
        at org.testng.internal.Invoker.invokeMethod(Invoker.java:661)
        at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:869)
        at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:1193)
        at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:126)
        at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:109)
        at org.testng.TestRunner.privateRun(TestRunner.java:744)
        at org.testng.TestRunner.run(TestRunner.java:602)
        at org.testng.SuiteRunner.runTest(SuiteRunner.java:380)
        at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:375)
        at org.testng.SuiteRunner.privateRun(SuiteRunner.java:340)
        at org.testng.SuiteRunner.run(SuiteRunner.java:289)
        at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
        at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:86)
        at org.testng.TestNG.runSuitesSequentially(TestNG.java:1301)
        at org.testng.TestNG.runSuitesLocally(TestNG.java:1226)
        at org.testng.TestNG.runSuites(TestNG.java:1144)
        at org.testng.TestNG.run(TestNG.java:1115)
        at org.apache.maven.surefire.testng.TestNGExecutor.run(TestNGExecutor.java:283)
        at org.apache.maven.surefire.testng.TestNGXmlTestSuite.execute(TestNGXmlTestSuite.java:75)
        at org.apache.maven.surefire.testng.TestNGProvider.invoke(TestNGProvider.java:120)
        at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:
373)
        at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:334)
        at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:119)
        at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:407)
]]>
            </full-stacktrace>
          </exception> <!-- java.lang.AssertionError -->
          <reporter-output>
          </reporter-output>
        </test-method> <!-- CanDoStuff -->
        <test-method status="PASS" signature="CanDivide()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="CanDivide" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanDivide -->
        <test-method status="PASS" signature="CanMultiplyX()[pri:0, instance:com.xpand.java.
CalcTest@36d4b5c]" name="CanMultiplyX" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-
06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
```

```
            <![CDATA[]]>
          </attribute> <!-- test -->
          <attribute name="requirement">
            <![CDATA[CALC-1235]]>
          </attribute> <!-- requirement -->
          <attribute name="labels">
            <![CDATA[core]]>
          </attribute> <!-- labels -->
        </attributes>
      </test-method> <!-- CanMultiplyX -->
      <test-method status="PASS" signature="CanSubtract()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="CanSubtract" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
        <reporter-output>
        </reporter-output>
        <attributes>
          <attribute name="test">
            <![CDATA[]]>
          </attribute> <!-- test -->
          <attribute name="requirement">
            <![CDATA[CALC-1235]]>
          </attribute> <!-- requirement -->
          <attribute name="labels">
            <![CDATA[core]]>
          </attribute> <!-- labels -->
        </attributes>
      </test-method> <!-- CanSubtract -->
      <test-method status="PASS" signature="tearDown()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="tearDown" is-config="true" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:
53:00Z">
        <reporter-output>
        </reporter-output>
      </test-method> <!-- tearDown -->
    </class> <!-- com.xpand.java.CalcTest -->
  </test> <!-- calculator -->
</suite> <!-- TestAll -->
</testng-results>
```

Example Request

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/testng?projectKey=XTP

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/testng?testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/testng?projectKey=XTP&testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/testng?projectKey=XTP&testPlanKey=XTP-12&revision=v2.1.0

Responses

200 OK : ***application/json*** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : **application/json** : *The Xray license is not valid.*

500 INTERNAL SERVER ERROR : **application/json** : *An internal error occurred when importing execution results.*

TestNG XML results Multipart

Xray provides another endpoint if you want to have control over the fields of newly created Test Executions and Tests. The endpoint accepts one XML file (the TestNG report) and two JSON files similar to the one Jira uses to create new issues, one with the fields of the Test Execution and the other with the fields of the Test. For more information about that JSON format, check the Jira documentation here. Xray will ignore the issue type and the summary fields on the Test JSON file since these are provided by the importation.

Import the execution results created with the TestNG XML output formatter. For more information please check the documentation about TestNG integration.

Request

Example

TestNG XML Report

```
<?xml version="1.0" encoding="UTF-8"?>
<testng-results skipped="0" failed="2" ignored="0" total="8" passed="6">
  <reporter-output>
  </reporter-output>
  <suite name="TestAll" duration-ms="33" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
    <groups>
    </groups>
    <test name="calculator" duration-ms="33" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:
00Z">
      <class name="com.xpand.java.CalcTest">
        <test-method status="PASS" signature="setUp()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]" name="
setUp" is-config="true" duration-ms="9" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
        </test-method> <!-- setUp -->
        <test-method status="PASS" signature="CanAddNumbers()[pri:0, instance:com.xpand.java.
CalcTest@36d4b5c]" name="CanAddNumbers" duration-ms="2" started-at="2018-03-06T11:53:00Z" finished-at="2018-
03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbers -->
        <test-method status="PASS" signature="CanAddNumbersFromGivenData(int, int, int)[pri:0, instance:com.
xpand.java.CalcTest@36d4b5c]" name="CanAddNumbersFromGivenData" duration-ms="0" started-at="2018-03-06T11:53:
00Z" data-provider="ValidDataProvider" finished-at="2018-03-06T11:53:00Z">
          <params>
            <param index="0">
              <value>
                <![CDATA[1]]>
              </value>
            </param>
            <param index="1">
              <value>
                <![CDATA[2]]>
              </value>
            </param>
```

```
            <param index="2">
              <value>
                <![CDATA[3]]>
              </value>
            </param>
          </params>
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbersFromGivenData -->
        <test-method status="FAIL" signature="CanAddNumbersFromGivenData(int, int, int)[pri:0, instance:com.
xpand.java.CalcTest@36d4b5c]" name="CanAddNumbersFromGivenData" duration-ms="1" started-at="2018-03-06T11:53:
00Z" data-provider="ValidDataProvider" finished-at="2018-03-06T11:53:00Z">
          <params>
            <param index="0">
              <value>
                <![CDATA[2]]>
              </value>
            </param>
            <param index="1">
              <value>
                <![CDATA[3]]>
              </value>
            </param>
            <param index="2">
              <value>
                <![CDATA[4]]>
              </value>
            </param>
          </params>
          <exception class="java.lang.AssertionError">
            <message>
              <![CDATA[expected [4] but found [5]]]>
            </message>
            <full-stacktrace>
              <![CDATA[java.lang.AssertionError: expected [4] but found [5]
        at org.testng.Assert.fail(Assert.java:93)
        at org.testng.Assert.failNotEquals(Assert.java:512)
        at org.testng.Assert.assertEqualsImpl(Assert.java:134)
        at org.testng.Assert.assertEquals(Assert.java:115)
        at org.testng.Assert.assertEquals(Assert.java:388)
        at org.testng.Assert.assertEquals(Assert.java:398)
        at com.xpand.java.CalcTest.CanAddNumbersFromGivenData(CalcTest.java:40)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:108)
        at org.testng.internal.Invoker.invokeMethod(Invoker.java:661)
        at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:869)
        at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:1193)
        at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:126)
        at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:109)
        at org.testng.TestRunner.privateRun(TestRunner.java:744)
        at org.testng.TestRunner.run(TestRunner.java:602)
        at org.testng.SuiteRunner.runTest(SuiteRunner.java:380)
        at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:375)
        at org.testng.SuiteRunner.privateRun(SuiteRunner.java:340)
        at org.testng.SuiteRunner.run(SuiteRunner.java:289)
        at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
        at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:86)
```

```
        at org.testng.TestNG.runSuitesSequentially(TestNG.java:1301)
        at org.testng.TestNG.runSuitesLocally(TestNG.java:1226)
        at org.testng.TestNG.runSuites(TestNG.java:1144)
        at org.testng.TestNG.run(TestNG.java:1115)
        at org.apache.maven.surefire.testng.TestNGExecutor.run(TestNGExecutor.java:283)
        at org.apache.maven.surefire.testng.TestNGXmlTestSuite.execute(TestNGXmlTestSuite.java:75)
        at org.apache.maven.surefire.testng.TestNGProvider.invoke(TestNGProvider.java:120)
        at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:
373)
        at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:334)
        at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:119)
        at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:407)
]]>
            </full-stacktrace>
          </exception> <!-- java.lang.AssertionError -->
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbersFromGivenData -->
        <test-method status="PASS" signature="CanAddNumbersFromGivenData(int, int, int)[pri:0, instance:com.
xpand.java.CalcTest@36d4b5c]" name="CanAddNumbersFromGivenData" duration-ms="0" started-at="2018-03-06T11:53:
00Z" data-provider="ValidDataProvider" finished-at="2018-03-06T11:53:00Z">
          <params>
            <param index="0">
              <value>
                <![CDATA[-1]]>
              </value>
            </param>
            <param index="1">
              <value>
                <![CDATA[1]]>
              </value>
            </param>
            <param index="2">
              <value>
                <![CDATA[0]]>
              </value>
            </param>
          </params>
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanAddNumbersFromGivenData -->
        <test-method status="FAIL" signature="CanDoStuff()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="CanDoStuff" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <exception class="java.lang.AssertionError">
            <message>
              <![CDATA[null]]>
            </message>
            <full-stacktrace>
              <![CDATA[java.lang.AssertionError: null
```

```
        at org.testng.Assert.fail(Assert.java:93)
        at org.testng.Assert.assertNotEquals(Assert.java:897)
        at org.testng.Assert.assertNotEquals(Assert.java:902)
        at com.xpand.java.CalcTest.CanDoStuff(CalcTest.java:86)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:108)
        at org.testng.internal.Invoker.invokeMethod(Invoker.java:661)
        at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:869)
        at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:1193)
        at org.testng.internal.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:126)
        at org.testng.internal.TestMethodWorker.run(TestMethodWorker.java:109)
        at org.testng.TestRunner.privateRun(TestRunner.java:744)
        at org.testng.TestRunner.run(TestRunner.java:602)
        at org.testng.SuiteRunner.runTest(SuiteRunner.java:380)
        at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:375)
        at org.testng.SuiteRunner.privateRun(SuiteRunner.java:340)
        at org.testng.SuiteRunner.run(SuiteRunner.java:289)
        at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)
        at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:86)
        at org.testng.TestNG.runSuitesSequentially(TestNG.java:1301)
        at org.testng.TestNG.runSuitesLocally(TestNG.java:1226)
        at org.testng.TestNG.runSuites(TestNG.java:1144)
        at org.testng.TestNG.run(TestNG.java:1115)
        at org.apache.maven.surefire.testng.TestNGExecutor.run(TestNGExecutor.java:283)
        at org.apache.maven.surefire.testng.TestNGXmlTestSuite.execute(TestNGXmlTestSuite.java:75)
        at org.apache.maven.surefire.testng.TestNGProvider.invoke(TestNGProvider.java:120)
        at org.apache.maven.surefire.booter.ForkedBooter.invokeProviderInSameClassLoader(ForkedBooter.java:
373)
        at org.apache.maven.surefire.booter.ForkedBooter.runSuitesInProcess(ForkedBooter.java:334)
        at org.apache.maven.surefire.booter.ForkedBooter.execute(ForkedBooter.java:119)
        at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:407)
]]>
            </full-stacktrace>
          </exception> <!-- java.lang.AssertionError -->
          <reporter-output>
          </reporter-output>
        </test-method> <!-- CanDoStuff -->
        <test-method status="PASS" signature="CanDivide()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="CanDivide" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanDivide -->
        <test-method status="PASS" signature="CanMultiplyX()[pri:0, instance:com.xpand.java.
CalcTest@36d4b5c]" name="CanMultiplyX" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-
06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
```

```
          </attributes>
        </test-method> <!-- CanMultiplyX -->
        <test-method status="PASS" signature="CanSubtract()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="CanSubtract" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:53:00Z">
          <reporter-output>
          </reporter-output>
          <attributes>
            <attribute name="test">
              <![CDATA[]]>
            </attribute> <!-- test -->
            <attribute name="requirement">
              <![CDATA[CALC-1235]]>
            </attribute> <!-- requirement -->
            <attribute name="labels">
              <![CDATA[core]]>
            </attribute> <!-- labels -->
          </attributes>
        </test-method> <!-- CanSubtract -->
        <test-method status="PASS" signature="tearDown()[pri:0, instance:com.xpand.java.CalcTest@36d4b5c]"
name="tearDown" is-config="true" duration-ms="0" started-at="2018-03-06T11:53:00Z" finished-at="2018-03-06T11:
53:00Z">
          <reporter-output>
          </reporter-output>
        </test-method> <!-- tearDown -->
      </class> <!-- com.xpand.java.CalcTest -->
    </test> <!-- calculator -->
  </suite> <!-- TestAll -->
</testng-results>
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Info JSON (Test)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "labels" : ["Server","TestNG"]
    }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.xml -F testInfo=@testIssueFields.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/testng/multipart

Responses

200 OK : ***application/json*** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : ***application/json*** *: The Xray license is not valid.*

500 INTERNAL SERVER ERROR : ***application/json*** *: An internal error occurred when importing execution results.*

Robot Framework XML results

After executing Robot Framework tests, you must import the outputted XML execution results to Jira using the following endpoint:

Import the execution results from Robot Framework XML output format. For more information please check the documentation about Robot Framework integration.

Request

QUERY PARAMETERS

parameter	type	description
`projectKey`	*String*	- key of the project where the Test Execution (if the **testExecKey** parameter wasn't provided) and the tests (if they aren't created yet) are going to be created.
testExecKey	String	- key of the Test Execution.
testPlanKey	String	- key of the Test Plan; if you specify the Test Plan, the Tests will be added automatically to the Test Plan if they're not part of it.
testEnvironments	String	- a string containing a list of test environments separated by ";"
revision	String	- source code and documentation version used in the test execution.
fixVersion	String	- the Fix Version associated with the test execution (it supports only one value).

Example

Robot Report XML

```
<?xml version="1.0" encoding="UTF-8"?>
<robot generated="20170220 14:18:54.562" generator="Robot 3.0.2 (Python 2.7.13 on win32)">
  <suite source="C:\Users\lmfv\Documents\Saco de Features\xray-1238\robot-example\robotframework-
webdemo\login_tests" id="s1" name="Login Tests">
    <suite source="C:\Users\lmfv\Documents\Saco de Features\xray-1238\robot-example\robotframework-
webdemo\login_tests\gherkin_login.robot" id="s1-s1" name="Gherkin Login">
      <test id="s1-s1-t1" name="Gherkin Valid Login">
```

```
        <kw name="Given browser is opened to login page">
          <kw name="Login Page Should Be Open" library="resource">
            <kw name="Title Should Be" library="Selenium2Library">
              <doc>Verifies that current page title equals `title`.</doc>
              <arguments>
                <arg>Log in - Your Company JIRA</arg>
              </arguments>
              <msg timestamp="20170220 14:19:07.693" level="INFO">Page title is 'Log in - Your Company JIRA'.<
/msg>
              <status status="PASS" endtime="20170220 14:19:07.693" starttime="20170220 14:19:07.158">
              </status>
            </kw>
            <status status="PASS" endtime="20170220 14:19:07.693" starttime="20170220 14:19:07.158">
            </status>
          </kw>
          <status status="PASS" endtime="20170220 14:19:07.693" starttime="20170220 14:18:55.937">
          </status>
        </kw>
        <kw name="When user &amp;quot;admin&amp;quot; logs in with password &amp;quot;password123&amp;quot;">
          <kw name="Input Username" library="resource">
            <arguments>
              <arg>${username}</arg>
            </arguments>
            <kw name="Input Text" library="Selenium2Library">
              <doc>Types the given `text` into text field identified by `locator`.</doc>
              <arguments>
                <arg>login-form-username</arg>
                <arg>${username}</arg>
              </arguments>
              <msg timestamp="20170220 14:19:07.696" level="INFO">Typing text 'admin' into text field 'login-
form-username'</msg>
              <status status="PASS" endtime="20170220 14:19:09.314" starttime="20170220 14:19:07.696">
              </status>
            </kw>
            <status status="PASS" endtime="20170220 14:19:09.314" starttime="20170220 14:19:07.695">
            </status>
          </kw>
          <kw name="Input Password" library="resource">
            <arguments>
              <arg>${password}</arg>
            </arguments>
            <kw name="Input Text" library="Selenium2Library">
              <doc>Types the given `text` into text field identified by `locator`.</doc>
              <arguments>
                <arg>login-form-password</arg>
                <arg>${password}</arg>
              </arguments>
              <msg timestamp="20170220 14:19:09.316" level="INFO">Typing text 'password123' into text field
'login-form-password'</msg>
              <status status="PASS" endtime="20170220 14:19:10.956" starttime="20170220 14:19:09.316">
              </status>
            </kw>
            <status status="PASS" endtime="20170220 14:19:10.956" starttime="20170220 14:19:09.315">
            </status>
          </kw>
          <kw name="Submit Credentials" library="resource">
            <kw name="Click Button" library="Selenium2Library">
              <doc>Clicks a button identified by `locator`.</doc>
              <arguments>
                <arg>login-form-submit</arg>
              </arguments>
              <msg timestamp="20170220 14:19:10.958" level="INFO">Clicking button 'login-form-submit'.</msg>
              <status status="PASS" endtime="20170220 14:19:17.476" starttime="20170220 14:19:10.958">
              </status>
            </kw>
            <status status="PASS" endtime="20170220 14:19:17.477" starttime="20170220 14:19:10.957">
            </status>
          </kw>
          <status status="PASS" endtime="20170220 14:19:17.478" starttime="20170220 14:19:07.695">
          </status>
        </kw>
```

```
        <kw name="Then welcome page should be open" library="resource">
          <kw name="Location Should Be" library="Selenium2Library">
            <doc>Verifies that current URL is exactly `url`.</doc>
            <arguments>
              <arg>${WELCOME URL}</arg>
            </arguments>
            <kw name="Capture Page Screenshot" library="Selenium2Library">
              <doc>Takes a screenshot of the current page and embeds it into the log.</doc>
              <msg timestamp="20170220 14:19:18.702" html="yes" level="INFO">&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;
&lt;td colspan="3"&gt;&lt;a href="selenium-screenshot-1.png"&gt;&lt;img src="selenium-screenshot-1.png"
width="800px"&gt;&lt;/a&gt;</msg>
              <status status="PASS" endtime="20170220 14:19:18.702" starttime="20170220 14:19:18.004">
              </status>
            </kw>
            <msg timestamp="20170220 14:19:18.705" level="FAIL">Location should have been 'http://localhost:
8080/secure/Dashboard.jspa' but was 'http://localhost:8080/login.jsp'</msg>
            <status status="FAIL" endtime="20170220 14:19:18.705" starttime="20170220 14:19:17.483">
            </status>
          </kw>
          <status status="FAIL" endtime="20170220 14:19:18.706" starttime="20170220 14:19:17.481">
          </status>
        </kw>
        <kw type="teardown" name="Close Browser" library="Selenium2Library">
          <doc>Closes the current browser.</doc>
          <status status="PASS" endtime="20170220 14:19:22.382" starttime="20170220 14:19:18.707">
          </status>
        </kw>
        <tags>
          <tag>WEB-1</tag>
          <tag>WEB-3</tag>
        </tags>
        <status status="FAIL" endtime="20170220 14:19:22.383" critical="yes" starttime="20170220 14:18:55.936"
>Location should have been 'http://localhost:8080/secure/Dashboard.jspa' but was 'http://localhost:8080/login.
jsp'</status>
      </test>
      <doc>A test suite with a single Gherkin style test.This test is functionally identical to the example
invalid_login.robot file.</doc>
      <status status="FAIL" endtime="20170220 14:19:22.397" starttime="20170220 14:18:54.670">
      </status>
    </suite>
    <status status="FAIL" endtime="20170220 14:22:12.549" starttime="20170220 14:18:54.567">
    </status>
  </suite>
</robot>
```


Example Request

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/robot?projectKey=XTP

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/robot?testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/robot?projectKey=XTP&testExecKey=XNP-23

curl -H "Content-Type: text/xml" -X POST -H "Authorization: Bearer $token" --data @"data.xml" https://xray.cloud.getxray.app/api/v2/import/execution/robot?projectKey=XTP&testPlanKey=XTP-12&revision=v2.1.0

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** Returns the error.

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

Robot Framework XML results Multipart

Xray provides another endpoint if you want to have control over the fields of newly created Test Executions and Tests. The endpoint accepts one XML file (the Robot report) and two JSON files similar to the one Jira uses to create new issues, one with the fields of the Test Execution and the other with the fields of the Test. For more information about that JSON format, check the Jira documentation here. Xray will ignore the issue type and the summary fields on the Test JSON file since these are provided by the importation.

Imports the execution results from Robot Framework XML output format. For more information please check the documentation about Robot Framework integration.

Request

Example

Robot Report XML

```
<?xml version="1.0" encoding="UTF-8"?>
<robot generated="20170220 14:18:54.562" generator="Robot 3.0.2 (Python 2.7.13 on win32)">
  <suite source="C:\Users\lmfv\Documents\Saco de Features\xray-1238\robot-example\robotframework-
webdemo\login_tests" id="s1" name="Login Tests">
    <suite source="C:\Users\lmfv\Documents\Saco de Features\xray-1238\robot-example\robotframework-
webdemo\login_tests\gherkin_login.robot" id="s1-s1" name="Gherkin Login">
      <test id="s1-s1-t1" name="Gherkin Valid Login">
        <kw name="Given browser is opened to login page">
          <kw name="Login Page Should Be Open" library="resource">
            <kw name="Title Should Be" library="Selenium2Library">
              <doc>Verifies that current page title equals `title`.</doc>
              <arguments>
                <arg>Log in - Your Company JIRA</arg>
              </arguments>
              <msg timestamp="20170220 14:19:07.693" level="INFO">Page title is 'Log in - Your Company JIRA'.<
/msg>
              <status status="PASS" endtime="20170220 14:19:07.693" starttime="20170220 14:19:07.158">
              </status>
            </kw>
            <status status="PASS" endtime="20170220 14:19:07.693" starttime="20170220 14:19:07.158">
            </status>
          </kw>
          <status status="PASS" endtime="20170220 14:19:07.693" starttime="20170220 14:18:55.937">
          </status>
        </kw>
        <kw name="When user &quot;admin&quot; logs in with password &quot;password123&quot;">
          <kw name="Input Username" library="resource">
            <arguments>
              <arg>${username}</arg>
            </arguments>
            <kw name="Input Text" library="Selenium2Library">
              <doc>Types the given `text` into text field identified by `locator`.</doc>
              <arguments>
                <arg>login-form-username</arg>
```

```
              <arg>${username}</arg>
            </arguments>
            <msg timestamp="20170220 14:19:07.696" level="INFO">Typing text 'admin' into text field 'login-
form-username'</msg>
            <status status="PASS" endtime="20170220 14:19:09.314" starttime="20170220 14:19:07.696">
            </status>
          </kw>
          <status status="PASS" endtime="20170220 14:19:09.314" starttime="20170220 14:19:07.695">
          </status>
        </kw>
        <kw name="Input Password" library="resource">
          <arguments>
            <arg>${password}</arg>
          </arguments>
          <kw name="Input Text" library="Selenium2Library">
            <doc>Types the given `text` into text field identified by `locator`.</doc>
            <arguments>
              <arg>login-form-password</arg>
              <arg>${password}</arg>
            </arguments>
            <msg timestamp="20170220 14:19:09.316" level="INFO">Typing text 'password123' into text field
'login-form-password'</msg>
            <status status="PASS" endtime="20170220 14:19:10.956" starttime="20170220 14:19:09.316">
            </status>
          </kw>
          <status status="PASS" endtime="20170220 14:19:10.956" starttime="20170220 14:19:09.315">
          </status>
        </kw>
        <kw name="Submit Credentials" library="resource">
          <kw name="Click Button" library="Selenium2Library">
            <doc>Clicks a button identified by `locator`.</doc>
            <arguments>
              <arg>login-form-submit</arg>
            </arguments>
            <msg timestamp="20170220 14:19:10.958" level="INFO">Clicking button 'login-form-submit'.</msg>
            <status status="PASS" endtime="20170220 14:19:17.476" starttime="20170220 14:19:10.958">
            </status>
          </kw>
          <status status="PASS" endtime="20170220 14:19:17.477" starttime="20170220 14:19:10.957">
          </status>
        </kw>
        <status status="PASS" endtime="20170220 14:19:17.478" starttime="20170220 14:19:07.695">
        </status>
      </kw>
      <kw name="Then welcome page should be open" library="resource">
        <kw name="Location Should Be" library="Selenium2Library">
          <doc>Verifies that current URL is exactly `url`.</doc>
          <arguments>
            <arg>${WELCOME URL}</arg>
          </arguments>
          <kw name="Capture Page Screenshot" library="Selenium2Library">
            <doc>Takes a screenshot of the current page and embeds it into the log.</doc>
            <msg timestamp="20170220 14:19:18.702" html="yes" level="INFO">&amp;lt;/td&amp;gt;&amp;lt;/tr&amp;gt;&amp;lt;tr&amp;gt;
&amp;lt;td colspan="3"&amp;gt;&amp;lt;a href="selenium-screenshot-1.png"&amp;gt;&amp;lt;img src="selenium-screenshot-1.png"
width="800px"&amp;gt;&amp;lt;/a&amp;gt;</msg>
            <status status="PASS" endtime="20170220 14:19:18.702" starttime="20170220 14:19:18.004">
            </status>
          </kw>
          <msg timestamp="20170220 14:19:18.705" level="FAIL">Location should have been 'http://localhost:
8080/secure/Dashboard.jspa' but was 'http://localhost:8080/login.jsp'</msg>
          <status status="FAIL" endtime="20170220 14:19:18.705" starttime="20170220 14:19:17.483">
          </status>
        </kw>
        <status status="FAIL" endtime="20170220 14:19:18.706" starttime="20170220 14:19:17.481">
        </status>
      </kw>
      <kw type="teardown" name="Close Browser" library="Selenium2Library">
        <doc>Closes the current browser.</doc>
        <status status="PASS" endtime="20170220 14:19:22.382" starttime="20170220 14:19:18.707">
        </status>
      </kw>
```

```
        <tags>
          <tag>WEB-1</tag>
          <tag>WEB-3</tag>
        </tags>
        <status status="FAIL" endtime="20170220 14:19:22.383" critical="yes" starttime="20170220 14:18:55.936"
>Location should have been 'http://localhost:8080/secure/Dashboard.jspa' but was 'http://localhost:8080/login.
jsp'</status>
      </test>
      <doc>A test suite with a single Gherkin style test.This test is functionally identical to the example
invalid_login.robot file.</doc>
      <status status="FAIL" endtime="20170220 14:19:22.397" starttime="20170220 14:18:54.670">
      </status>
    </suite>
    <status status="FAIL" endtime="20170220 14:22:12.549" starttime="20170220 14:18:54.567">
    </status>
  </suite>
</robot>
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "summary": "Brand new Test execution",
        "issuetype": {
            "id": "10007"
        },
        "components" : [
            {
            "name":"Interface"
            },
            {
            "name":"Core"
            }
        ]
    },
        "xrayFields": {
                "testPlanKey": "DEMO-15",
                "environments": ["Chrome", "Windows"]
        }
}
```

Info JSON (Test)

```
{
    "fields": {
        "project": {
            "id": "10402"
        },
        "labels" : ["Server","Robot"]
    }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.xml -F testInfo=@testIssueFields.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/robot/multipart

Responses

200 OK : ***application/json*** *: Successful. The results were successfully imported to Jira.*

Example Output

```
{
  "id": "10200",
  "key": "DEMO-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : ***application/json :*** *Returns the error.*

401 UNAUTHORIZED : ***application/json*** *: The Xray license is not valid.*

500 INTERNAL SERVER ERROR : ***application/json*** *: An internal error occurred when importing execution results.*

Behave JSON results

After executing Behave features, you must import the outputted JSON execution results to Jira using the following endpoint:

Import the execution results created with the Behave JSON output formatter.

Request

Example

Example Input

```
[
{
  "elements": [
    {
      "keyword": "Background",
      "location": "features/tutorial09_background.feature:3",
      "name": "Ninja fight setup",
      "steps": [
        {
          "keyword": "Given",
          "location": "features/tutorial09_background.feature:4",
          "name": "the ninja encounters another opponent",
          "step_type": "given"
        }
      ],
      "type": "background"
    },
    {
      "keyword": "Scenario",
      "location": "features/tutorial09_background.feature:7",
      "name": "Weaker opponent",
      "status": "passed",
      "steps": [
        {
          "keyword": "Given",
          "location": "features/tutorial09_background.feature:4",
          "match": {
            "arguments": [],
            "location": "features/steps/step_tutorial02.py:78"
          },
          "name": "the ninja encounters another opponent",
          "result": {
            "duration": 0.0007519721984863281,
```

```
          "status": "passed"
        },
        "step_type": "given"
      },
      {
        "keyword": "Given",
        "location": "features/tutorial09_background.feature:8",
        "match": {
          "arguments": [
            {
              "name": "achievement_level",
              "value": "third level black-belt"
            }
          ],
          "location": "features/steps/step_tutorial02.py:57"
        },
        "name": "the ninja has a third level black-belt",
        "result": {
          "duration": 8.988380432128906e-05,
          "status": "passed"
        },
        "step_type": "given"
      },
      {
        "keyword": "When",
        "location": "features/tutorial09_background.feature:9",
        "match": {
          "arguments": [
            {
              "name": "opponent_role",
              "value": "samurai"
            }
          ],
          "location": "features/steps/step_tutorial02.py:61"
        },
        "name": "attacked by a samurai",
        "result": {
          "duration": 6.29425048828125e-05,
          "status": "passed"
        },
        "step_type": "when"
      },
      {
        "keyword": "Then",
        "location": "features/tutorial09_background.feature:10",
        "match": {
          "arguments": [
            {
              "name": "reaction",
              "value": "engage the opponent"
            }
          ],
          "location": "features/steps/step_tutorial02.py:69"
        },
        "name": "the ninja should engage the opponent",
        "result": {
          "duration": 7.295608520507812e-05,
          "status": "passed"
        },
        "step_type": "then"
      }
    ],
    "tags": [
      "TEST_DEMO-3",
      "TESTSET_DEMO-6",
      "REQ_DEMO-7"
    ],
    "type": "scenario"
  },
  {
    "keyword": "Scenario",
```

```
"location": "features/tutorial09_background.feature:13",
"name": "Stronger opponent",
"status": "passed",
"steps": [
  {
    "keyword": "Given",
    "location": "features/tutorial09_background.feature:4",
    "match": {
      "arguments": [],
      "location": "features/steps/step_tutorial02.py:78"
    },
    "name": "the ninja encounters another opponent",
    "result": {
      "duration": 8.487701416015625e-05,
      "status": "passed"
    },
    "step_type": "given"
  },
  {
    "keyword": "Given",
    "location": "features/tutorial09_background.feature:14",
    "match": {
      "arguments": [
        {
          "name": "achievement_level",
          "value": "second level black-belt"
        }
      ],
      "location": "features/steps/step_tutorial02.py:57"
    },
    "name": "the ninja has a second level black-belt",
    "result": {
      "duration": 8.130073547363281e-05,
      "status": "passed"
    },
    "step_type": "given"
  },
  {
    "keyword": "When",
    "location": "features/tutorial09_background.feature:15",
    "match": {
      "arguments": [
        {
          "name": "opponent",
          "value": "Chuck Norris"
        }
      ],
      "location": "features/steps/step_tutorial02.py:65"
    },
    "name": "attacked by Chuck Norris",
    "result": {
      "duration": 6.127357482910156e-05,
      "status": "passed"
    },
    "step_type": "when"
  },
  {
    "keyword": "Then",
    "location": "features/tutorial09_background.feature:16",
    "match": {
      "arguments": [
        {
          "name": "reaction",
          "value": "run for his life"
        }
      ],
      "location": "features/steps/step_tutorial02.py:69"
    },
    "name": "the ninja should run for his life",
    "result": {
      "duration": 8.821487426757812e-05,
```

```
            "status": "passed"
          },
          "step_type": "then"
        }
      ],
      "tags": [
        "TEST_DEMO-4",
        "TESTSET_DEMO-6",
        "REQ_DEMO-7"
      ],
      "type": "scenario"
    }
  ],
  "keyword": "Feature",
  "location": "features/tutorial09_background.feature:1",
  "name": "Using Background -- Fight or Flight",
  "status": "passed",
  "tags": []
}
]
```

Example Request

curl -H "Content-Type: application/json" -X POST -H "Authorization: Bearer $token" --data @"data.json" https://xray.cloud.getxray.app/api/v2/import/execution/behave

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "XNP-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** No execution results were provided**.**

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.

Behave JSON results Multipart

Xray provides another endpoint if you want to create new Test Executions and have control over newly-created Test Execution fields. It allows you to send two JSON files, the normal Behave result JSON and a JSON similar to the one Jira uses to create new issues. For more information about that second format, check the Jira documentation here.

Import the execution results created with the Behave JSON output formatter.

Request

Example

Example Input

```
[
{
```

```
"elements": [
  {
    "keyword": "Background",
    "location": "features/tutorial09_background.feature:3",
    "name": "Ninja fight setup",
    "steps": [
      {
        "keyword": "Given",
        "location": "features/tutorial09_background.feature:4",
        "name": "the ninja encounters another opponent",
        "step_type": "given"
      }
    ],
    "type": "background"
  },
  {
    "keyword": "Scenario",
    "location": "features/tutorial09_background.feature:7",
    "name": "Weaker opponent",
    "status": "passed",
    "steps": [
      {
        "keyword": "Given",
        "location": "features/tutorial09_background.feature:4",
        "match": {
          "arguments": [],
          "location": "features/steps/step_tutorial02.py:78"
        },
        "name": "the ninja encounters another opponent",
        "result": {
          "duration": 0.0007519721984863281,
          "status": "passed"
        },
        "step_type": "given"
      },
      {
        "keyword": "Given",
        "location": "features/tutorial09_background.feature:8",
        "match": {
          "arguments": [
            {
              "name": "achievement_level",
              "value": "third level black-belt"
            }
          ],
          "location": "features/steps/step_tutorial02.py:57"
        },
        "name": "the ninja has a third level black-belt",
        "result": {
          "duration": 8.988380432128906e-05,
          "status": "passed"
        },
        "step_type": "given"
      },
      {
        "keyword": "When",
        "location": "features/tutorial09_background.feature:9",
        "match": {
          "arguments": [
            {
              "name": "opponent_role",
              "value": "samurai"
            }
          ],
          "location": "features/steps/step_tutorial02.py:61"
        },
        "name": "attacked by a samurai",
        "result": {
          "duration": 6.29425048828125e-05,
          "status": "passed"
        },
```

```
        "step_type": "when"
      },
      {
        "keyword": "Then",
        "location": "features/tutorial09_background.feature:10",
        "match": {
          "arguments": [
            {
              "name": "reaction",
              "value": "engage the opponent"
            }
          ],
          "location": "features/steps/step_tutorial02.py:69"
        },
        "name": "the ninja should engage the opponent",
        "result": {
          "duration": 7.295608520507812e-05,
          "status": "passed"
        },
        "step_type": "then"
      }
    ],
    "tags": [
      "TEST_DEMO-3",
      "TESTSET_DEMO-6",
      "REQ_DEMO-7"
    ],
    "type": "scenario"
  },
  {
    "keyword": "Scenario",
    "location": "features/tutorial09_background.feature:13",
    "name": "Stronger opponent",
    "status": "passed",
    "steps": [
      {
        "keyword": "Given",
        "location": "features/tutorial09_background.feature:4",
        "match": {
          "arguments": [],
          "location": "features/steps/step_tutorial02.py:78"
        },
        "name": "the ninja encounters another opponent",
        "result": {
          "duration": 8.487701416015625e-05,
          "status": "passed"
        },
        "step_type": "given"
      },
      {
        "keyword": "Given",
        "location": "features/tutorial09_background.feature:14",
        "match": {
          "arguments": [
            {
              "name": "achievement_level",
              "value": "second level black-belt"
            }
          ],
          "location": "features/steps/step_tutorial02.py:57"
        },
        "name": "the ninja has a second level black-belt",
        "result": {
          "duration": 8.130073547363281e-05,
          "status": "passed"
        },
        "step_type": "given"
      },
      {
        "keyword": "When",
        "location": "features/tutorial09_background.feature:15",
```

```
          "match": {
            "arguments": [
              {
                "name": "opponent",
                "value": "Chuck Norris"
              }
            ],
            "location": "features/steps/step_tutorial02.py:65"
          },
          "name": "attacked by Chuck Norris",
          "result": {
            "duration": 6.127357482910156e-05,
            "status": "passed"
          },
          "step_type": "when"
        },
        {
          "keyword": "Then",
          "location": "features/tutorial09_background.feature:16",
          "match": {
            "arguments": [
              {
                "name": "reaction",
                "value": "run for his life"
              }
            ],
            "location": "features/steps/step_tutorial02.py:69"
          },
          "name": "the ninja should run for his life",
          "result": {
            "duration": 8.821487426757812e-05,
            "status": "passed"
          },
          "step_type": "then"
        }
      ],
      "tags": [
        "TEST_DEMO-4",
        "TESTSET_DEMO-6",
        "REQ_DEMO-7"
      ],
      "type": "scenario"
    }
  ],
  "keyword": "Feature",
  "location": "features/tutorial09_background.feature:1",
  "name": "Using Background -- Fight or Flight",
  "status": "passed",
  "tags": []
}
]
```

Info JSON (Test Execution)

```
{
    "fields": {
        "project": {
            "key": "DEMO"
        },
        "summary": "Brand new Test execution (behave multipart)",
        "issuetype": {
            "id": "10008"
        }
    },
    "xrayFields": {
        "testPlanKey": "DEMO-9"
    }
}
```

Example Request

curl -H "Content-Type: multipart/form-data" -X POST -F info=@issueFields.json -F results=@results.json -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/execution/behave/multipart

Responses

200 OK : **application/json** : Successful. The results were successfully imported to Jira.

Example Output

```
{
  "id": "10200",
  "key": "DEMO-24",
  "self": "https://www.example.com/rest/api/2/issue/10200"
}
```

400 BAD_REQUEST : **application/json :** No execution results were provided**.**

401 UNAUTHORIZED : **application/json** : The Xray license is not valid.

500 INTERNAL SERVER ERROR : **application/json** : An internal error occurred when importing execution results.