

Importing Tests - REST v2

There are two endpoints regarding importing tests in bulk.

Import Tests	/api/v2/import/test/bulk
Check import job status	/api/v2/import/test/bulk/{jobId}/status

The first endpoint is an asynchronous call to import tests. Whenever this endpoint is called, a job is queued to import the tests in the input and its id is returned. The second endpoint can be used to check the status of this job.

Import Tests

When importing tests, you can use the following endpoint:

Create a job to import tests.

Request

PATH PARAMETERS

parameter	type	description
project	String	- key or id of the project where the tests without project information are to be imported into.

In the issue information in the input, the following fields can be used:

- **xray_issue_type**: The type of the issue being imported. Accepted values are "test", "testset", "test_set" or "test set" (case insensitive). If not provided will default to test.
- **testtype (or xray_testtype)**: This field determines the test type of the corresponding test, this field can be filled with the option of any test field that is present in the project in which the test will be created. (this field is required)
- **steps (or xray_steps)**: It holds the steps information for a test with a test type of the Steps kind.
- **gherkin_def (or xray_gherkin_def)**: Gherkin definition for tests with a test type of the Gherkin kind.
- **unstructured_def (or xray_unstructured_def)**: Generic definition for tests with a test type of Unstructured kind.
- **xray_test_repository_folder**: Test Repository path to where the respective test is to be moved into.
- **xray_test_sets**: A list of test sets to which the test is to be added. Each value should be the issue key of an existing test set or the id of a test set also being imported ("xray_id" field).
- **xray_id**: A unique identifier for a test set in the input. Used to add tests to the test set (see "xray_test_sets" field). (required for test sets)

Along with these fields, JIRA fields and update information can be added as used in the bulk create issues of the JIRA REST API. The JIRA fields information may be expressed together with the Xray fields or inside the *fields* field.



JIRA fields

The JIRA fields and update structure must follow the one used in the [Create Issues of the JIRA Cloud API](#)

If project information is omitted in an issue, the project defined as a path parameter in the endpoint is used as the destination project. If no project is set as a path parameter, an error will occur importing the corresponding issue and any other issue without project information.

Example 1: Tests

Example Input

```

[
    {
        "testtype": "Cucumber",
        "fields": {
            "summary": "This tests the addition of 2 numbers.",
            "project": { "key": "CALC" }
        },
        "gherkin_def": "Given I have entered <input_1> into the calculator\nAnd I have entered  
<input_2> into the calculator\nWhen I press <button>\nThen the result should be <output> on the screen\n\nExamples:\n | input_1 | input_2 | button | output |\n 2       | 5       | add    | 7       |\n add     | 41      |",
        "xray_test_sets": ["ts1"]
    },
    {
        "testtype": "Manual",
        "fields": {
            "summary": "This tests the Subtraction of two numbers.",
            "project": { "key": "CALC" }
        },
        "update": {
            "issuelinks": [
                {
                    "add": {
                        "type": {
                            "name": "Test"
                        },
                        "outwardIssue": {
                            "key": "CALC-123"
                        }
                    }
                }
            ]
        },
        "steps": [
            {
                "action": "I choose the operation of the calculator *Subtraction*",
                "data": "",
                "result": "The operation must appear selected."
            },
            {
                "action": "I enter the input into the calculator",
                "data": "I1: 5\nI2: 2",
                "result": ""
            },
            {
                "action": "I press the *Calculate* button",
                "result": "The result *3* should be displayed in the screen, on the right of  
the \"=\" sign."
            }
        ],
        "xray_test_sets": ["ts1", "CALC-100"]
    },
    {
        "testtype": "Generic",
        "fields": {
            "summary": "CanMultiply",
            "project": { "key": "CALC" }
        },
        "unstructured_def": "x.CalculatorTests.CanMultiply",
        "xray_test_repository_folder": "Generic Tests/Arithmetic Tests"
    },
    {
        "xray_issue_type": "testset",
        "xray_id": "ts1",
        "fields": {
            "summary": "Calculator tests"
        }
    }
]

```



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/test/bulk
```

Responses

200 OK : **application/json** : Successful. The import job was created successfully and the Job Id is provided.

Example Output

```
{
  "jobId": "34a4106b1d0948d1aae1170cc8df3bb4"
}
```

400 BAD_REQUEST : **application/json** : No import test job was created. The content may be in the wrong format or a job might already be in progress. Each user can only have one import job running at each given time. So if a job is running and the user tries to start another one, this will result in a 400 response. Also there is a maximum of 1000 tests that can be imported each time, if the input contains more tests than this, the request will also result in a 400 response.

Example Output

```
{
  "error": "A job to import tests is already in progress (id: 34a4106b1d0948d1aae1170cc8df3bb4)."
}
```

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

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

Check Import Job Status

When checking the status of an import test job, you can use the following endpoint:

Check an import test job status.

Request

PARAMETERS

parameter	type	description
jobId	String	- id of the import tests job.

When a job is created its status becomes available using this endpoint, it will remain so for approximately one day, after that the job status is removed.

The *status* field has one of the following values:

- **pending:** the job was created but is not running yet.
- **working:** the job is currently running.
- **failed:** the job ended in failure.
- **successful:** all tests were successfully imported and the job ended.
- **partially_successful:** some tests were imported successfully and some weren't and the job has finished.
- **unsuccessful:** no tests were imported and job has finished.

When the job ends, in the status the result will be returned. This result has 2 sections, the *errors* section and the *issues* section, the first one has all the tests that weren't imported along with the errors that prevented them from being imported, the second section has the information of the issues that were successfully imported along with the JIRA issue id, key and self URL. In both section the *elementNumber* field indicates the index of the test it refers to in the input.



Example Request

```
curl -H "Content-Type: application/json" -X GET -H "Authorization: Bearer $token" https://xray.cloud.getxray.app/api/v2/import/test/bulk/34a4106b1d0948d1aae1170cc8df3bb4/status
```

Responses

200 OK : **application/json** : Successful. The job status is available.

Example 1: Job is still working

Example Output

```
{
  "status": "working",
  "progress": [
    "[Tue, 07 Aug 2018 13:54:37 GMT] Preprocessing the information of the 3 tests to import.",
    "[Tue, 07 Aug 2018 13:54:37 GMT] Creating 3 issue(s) in JIRA.",
    "[Tue, 07 Aug 2018 13:54:39 GMT] Saving additional information of 2 issue(s).",
  ],
  "progressValue": 80
}
```

Example 2: Job ended in partial success

Example Output 2

```
{
  "status": "partially_successful",
  "result": {
    "errors": [
      {
        "elementNumber": 1,
        "errors": {
          "description": "Operation value must be a string"
        }
      }
    ],
    "issues": [
      {
        "elementNumber": 0,
        "id": "12603",
        "key": "CALC-1",
        "self": "https://xpandsdcv.atlassian.net/rest/api/2/issue/12603"
      },
      {
        "elementNumber": 2,
        "id": "12604",
        "key": "CALC-2",
        "self": "https://xpandsdcv.atlassian.net/rest/api/2/issue/12604"
      }
    ]
  }
}
```

404 NOT_FOUND : **application/json** : Job was not found with the given jobId.

Example Output

```
{
  "error": "job not found."
}
```

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

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