Tests - REST

You can manage Tests directly through the Jira REST API.

- Creating and Editing Tests - Jira REST API
- Exporting Tests
- Getting all Tests statuses
- Exporting Test Runs
- Exporting Test Pre-Conditions
- Exporting Test Sets
- Exporting Test Executions
- Exporting Test Plans

- Test Steps - REST

Creating and Editing Tests - Jira REST API

Xray Test issues can be created and edited though the Jira REST API.

> **Learn more**
>
> https://developer.atlassian.com/display/JIRADEV/JIRA+REST+API+Example+-+Create+Issue
>
> https://developer.atlassian.com/display/JIRADEV/JIRA+REST+API+Example+-+Edit+issues

All Test issue custom fields provided by Xray can be modified using the Jira REST API:

- **Test Type** *(default Jira select field)*
- **Cucumber Test Type** *(default Jira select field)*
- **Cucumber Scenario** *(default Jira text field)*
- **Manual Test Steps** *(JSON format; example below)*

> The custom field IDs can be obtained using the Jira REST API Browser tool included in Jira. Each ID is of the form "**customfield_ID**".
>
> Another option, which does not require Jira administration rights, is to invoke the "Get edit issue meta" in an existing issue (e.g., in a Test issue) as mentioned here.
>
> Example: GET http://yourserver/rest/api/2/issue/CALC-1/editmeta

"Manual Test Steps" custom field value format

```
{
   "fields": {
       "customfield_10004": {
                       "steps": [
                               {
                                       "index": 0,
                                       "step": "Step 1",
                                       "data": "input Data 1",
                                       "result": "Excepted result 1"
                               },
                               {
                                       "index": 1,
                                       "step": "Step 2",
                                       "data": "input Data 2",
                                       "result": "Excepted result 2"
                               },
                               {
                                       "index": 2,
                                       "step": "Step 3",
                                       "data": "input Data 3",
                                       "result": "Excepted result 3"
                               },
                                {
                                       "index": 3,
                                       "step": "Step 4",
                                       "data": "input Data 4",
                                       "result": "Excepted result 4"
                               }
                       ]
               }
   }
}
```

Below is a full Test creation example using Jira's REST API endpoint for creating issues.

Creates a Test in a project.

Request

Example Input for Manual Test

```
{
    "fields": {
       "project":
       {
          "key": "ABC"
       },
       "summary": "Sum of two numbers",
       "description": "example of manual test",
       "issuetype": {
          "name": "Test"
       },

       "customfield_10200": { "value": "Manual" },
       "customfield_10004": {
            "steps": [
                {
                    "index": 0,
                    "step": "Step 1",
                    "data": "input Data 1",
                    "result": "Excepted result 1"
                },
                {
                    "index": 1,
                    "step": "Step 2",
                    "data": "input Data 2",
                    "result": "Excepted result 2"
                },
                {
                    "index": 2,
                    "step": "Step 3",
                    "data": "input Data 3",
                    "result": "Excepted result 3"
                },
                {
                    "index": 3,
                    "step": "Step 4",
                    "data": "input Data 4",
                    "result": "Excepted result 4"
                }
            ]
        }

   }
}
```

where:

- "customfield_10200" corresponds to the "Test Type" custom field (please check the correct customfield id for your Jira instance)
- "customfield_10004" corresponds to the "Manual Test Steps" custom field (please check the correct customfield id for your Jira instance)

Example Input for automated test (Cucumber Scenario)

```
{
    "fields": {
       "project":
       {
          "key": "ABC"
       },
       "summary": "Sum of two numbers",
       "description": "example of cucumber automated test - Scenario",
       "issuetype": {
          "name": "Test"
       },


       "customfield_10200": { "value": "Cucumber" },
       "customfield_10201": { "value": "Scenario" },
       "customfield_10202": "Given I have a calculator\nWhen I press 1\nAnd I press +\nAnd I press 2\nAnd I
press =\nThen I should see 3"

   }
}
```

where:

- "customfield_10200" corresponds to the "Test Type" custom field (please check the correct customfield id for your Jira instance)
- "customfield_10201" corresponds to the "Cucumber Test Type" custom field (please check the correct customfield id for your Jira instance)
- "customfield_10202" corresponds to the "Cucumber Scenario" custom field (please check the correct customfield id for your Jira instance)

Example Input for automated test (Generic)

```
{
    "fields": {
       "project":
       {
          "key": "ABC"
       },
       "summary": "Sum of two number",
       "description": "example of generic test",
       "issuetype": {
          "name": "Test"
       },


       "customfield_10200": { "value": "Generic" },
       "customfield_10203": "sum_script.sh"

   }
}
```

where:

- `"customfield_10200" corresponds to the "Test Type" custom field (please check the correct customfield id for your Jira instance)`
- `"customfield_10203" corresponds to the "Generic Test Definition" custom field (please check the correct customfield id for your Jira instance)`

Example Request

curl -H "Content-Type: application/json" -X POST --data @test.json -u admin:admin http://yourjiraserver/rest/api/2/issue

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output

```
{
    "id": "10700",
    "key": "ABC-53",
    "self": "http://127.0.0.1:8080/rest/api/2/issue/10700"
}
```

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

401 UNAUTHORIZED : **text/plain** : Unauthorized request.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred.

Exporting Tests

To export tests to JSON, you need to specify the keys, the ID of the filter or JQL query of the issues you want to export. At least one query parameter has to be specified, but all 3 can be sent at the same time.

Returns a json with the exported tests.

Request

QUERY PARAMETERS

parameter	type	description
keys	*String*	- list of keys of the tests separated by ';'.
filter	Long	- id of the filter.
jql	String	- jql query.

Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test?keys=TEST-123;TEST-321

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test?filter=12030

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test?jql=project%20=%2011000%20and%20issuetype%20=%207

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test?keys=TEST-123;TEST-321&filter=12030&jql=project%20=%2011000%20and%20issuetype%20=%207

Note: It's necessary to URl escape the JQL query.

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output


```
[
   {
      "key":"PER-1098",
      "self":"http://yourserver/rest/api/2/issue/15798",
      "reporter":"admin",
      "assignee":"admin",
      "precondition":[
         {
            "preconditionKey":"PER-1232",
            "self":"http://yourserver/rest/api/2/issue/16000",
            "reporter":"admin",
            "assignee":"admin",
            "type":"Automated[Cucumber]",
            "condition":"Condition example"
         }
      ],
      "type":"Automated[Cucumber]",
          "status":"TODO",
      "definition":"
Given I have entered <input_1> into the calculator\nAnd I have entered <input_2> into the calculator\nWhen I p
ress <button>\nThen the result should be <output> on the screen\n\nExamples: \n    | input_1 | input_2 | butto
n | output |\n    | 20      | 30      | add    | 50     |\n    | 2       | 5       | add    | 7      |\n    |
0       | 40      | add    | 40     |"
   },
   {
      "key":"PER-1099",
      "self":"http://yourserver/rest/api/2/issue/15799",
      "reporter":"admin",
      "assignee":"admin",
      "precondition":[

      ],
      "type":"Automated[Generic]",
          "status":"FAIL",
      "definition":"Test definition example"
   },
   {
      "key":"PER-1233",
      "self":"http://yourserver/rest/api/2/issue/16001",
      "reporter":"admin",
      "assignee":"admin",
      "precondition":[

      ],
      "type":"Manual",
          "status":"PASS",
      "definition":{
```

```
         "steps":[
            {
               "id":10940,
               "index":1,
               "step":{
                  "raw":"step1",
                  "rendered":"<p>step1</p>"
               },
               "data":{
                  "raw":"data1",
                  "rendered":"<p>data1</p>"
               },
               "result":{
                  "raw":"result1",
                  "rendered":"<p>result1</p>"
               },
               "attachments":[

               ]
            },
            {
               "id":10941,
               "index":2,
               "step":{
                  "raw":"step2",
                  "rendered":"<p>step2</p>"
               },
               "data":{
                  "raw":"data2",
                  "rendered":"<p>data2</p>"
               },
               "result":{
                  "raw":"result2",
                  "rendered":"<p>result2</p>"
               },
               "attachments":[

               ]
            }
         ]
      }
   }
]
```


400 BAD_REQUEST : **text/plain** : Returns the error.

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test steps.

Getting all Tests statuses

It is possible to get all Test Statuses available in Xray sorted by rank.

Returns a json with the test statuses.

Request

Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/settings/teststatuses

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output

```
[
 {
   "id": 1,
   "rank": 1,
   "name": "PASS",
   "description": "The test run has passed",
   "color": "#95C160",
   "final": true,
   "requirementStatusName": "OK"
 },
 ...
 {
   "id": 10,
   "rank": 10,
   "name": "BLOCKED",
   "description": "The test run is blocked",
   "color": "#111111",
   "final": false,
   "requirementStatusName": "NOTRUN"
 }

]
```

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test steps.

Exporting Test Runs

To export the test runs of a Test, you need to specify the key of the Test you wish to export the test runs from. You can filter the Test runs by Test Environment.

Returns a json with the exported test runs.

Request

PATH PARAMETERS

parameter	type	description
key	*String*	- Test Key

QUERY PARAMETERS

parameter	type	description
testEnvironments	*String*	- Test execution environments separated by ',' if the environment contains a comma escape it with '\'

Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test/{key}/testruns

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test/{key}/testruns?testEnvironments=IOS,Android

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output


```
[
   {
      "id":1587,
      "status":"FAIL",
          "testKey":"CALC-12",
          "testExecKey":"CALC-13",
      "assignee":"admin",
      "executedBy":"admin",
      "startedOn":"2016-10-11T17:15:05+01:00",
      "finishedOn":"2016-10-24T14:58:34+01:00",
          "duration":1115009068,
      "defects":[
         {
            "id":12410,
            "key":"PER-174",
            "summary":"Test Bug",
            "status":"Open"
         }
      ],
      "evidences":[
         {
            "id":383,
            "fileName":"result.json",
            "fileSize":"15 kB",
            "created":"Today 5:41 PM",
            "author":"admin",
            "fileURL":"http://JIRASERVER/plugins/servlet/raven/attachment/383/result.json"
         }
      ],
      "scenarioOutline":"
Given I have entered <input_1> into the calculator And I have entered <input_2> into the calculator When I pre
ss <button> Then the result should be <output> on the screen Examples: | input_1 | input_2 | button | output |
 | 20 | 30 | add | 50 | | 2 | 5 | add | 7 | | 0 | 40 | add | 40 |",
      "examples":[
         {
            "id":1,
            "rank":0,
            "values":[
               "input_1",
               "input_2",
               "button",
               "output"
            ],
            "status":"PASS"
         },
         {
            "id":2,
            "rank":1,
            "values":[
```

```
                "20",
                "30",
                "add",
                "50"
             ],
             "status":"FAIL"
          },
          {
             "id":3,
             "rank":2,
             "values":[
                "2",
                "5",
                "add",
                "7"
             ],
             "status":"PASS"
          },
          {
             "id":4,
             "rank":3,
             "values":[
                "0",
                "40",
                "add",
                "40"
             ],
             "status":"PASS"
          }
       ]
    },
    {
       "id":1588,
       "status":"TODO",
           "testKey":"CALC-12",
           "testExecKey":"CALC-13",
       "assignee":"admin",
       "defects":[
          {
             "id":12410,
             "key":"PER-174",
             "summary":"Test Bug",
             "status":"Open"
          }
       ],
       "evidences":[
          {
             "id":383,
             "fileName":"result.json",
             "fileSize":"15 kB",
             "created":"Today 5:41 PM",
             "author":"admin",
             "fileURL":"http://JIRASERVER/plugins/servlet/raven/attachment/383/result.json"
          }
       ],
       "scenarioOutline":"
Given I have entered <input_1> into the calculator And I have entered <input_2> into the calculator When I pre
ss <button> Then the result should be <output> on the screen Examples: | input_1 | input_2 | button | output |
 | 20 | 30 | add | 50 | | 2 | 5 | add | 7 | | 0 | 40 | add | 40 |",
       "examples":[]
    }
]
```


400 BAD_REQUEST : **text/plain** : Returns the error.

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test steps.

Exporting Test Pre-Conditions

To export the pre-conditions of a Test, you need to specify the key of the Test you wish to export the test pre-conditions from

Returns a json with the test pre-conditions of a given test.

Request

PATH PARAMETERS

parameter	type	description
key	*String*	- key of the test.

Example Request
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test/FLYTOMOON-76/preconditions

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output

```
[
    {
        "id": 12602,
        "rank": 1,
        "key": "FLYTOMOON-7",
        "self": "http://localhost:8080/rest/api/2/issue/12602",
        "reporter": "john",
        "assignee": "bmpc",
        "type": "Manual",
        "condition": "Fuel the rockets before launch"
    },
    {
        "id": 13409,
        "rank": 2,
        "key": "FLYTOMOON-77",
        "self": "http://localhost:8080/rest/api/2/issue/13409",
        "reporter": "admin",
        "assignee": "admin",
        "type": "Manual",
        "condition": "The PC must be turned on"
    },
    {
        "id": 12718,
        "rank": 3,
        "key": "FLYTOMOON-29",
        "self": "http://localhost:8080/rest/api/2/issue/12718",
        "reporter": "admin",
        "assignee": "admin",
        "type": "Manual",
        "condition": "Ligar o computador.."
    }
]
```

400 BAD_REQUEST : **text/plain** : Returns the error.

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test pre-conditions.

Exporting Test Sets

To export the test sets of a Test, you need to specify the key of the Test you wish to export the test sets from

Returns a json with the exported test sets.

Request

PATH PARAMETERS

parameter	type	description
key	*String*	- Test Key

Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test/{key}/testsets

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output

```
[
   {
      "id":13602,
      "key":"CALC-50",
      "summary":"Test Set for all tests",
      "self":"http://JIRASERVER/testJira/rest/api/2/issue/CALC-50",
          "environments" : [
                "IOS",
                "Android"
          ]
   },
   {
      "id":13600,
      "key":"CALC-51",
      "summary":"Test Set of v2.0",
      "self":"http://JIRASERVER/testJira/rest/api/2/issue/CALC-51",
          "environments" : [
                "IOS",
                "Android"
          ]
   }
]
```

400 BAD_REQUEST : **text/plain** : Returns the error.

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test steps.

Exporting Test Executions

To export the test executions of a Test, you need to specify the key of the Test you wish to export the test executions from

Returns a json with the exported test executions.

Request

PATH PARAMETERS

parameter	type	description
key	*String*	- Test Key

Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test/{key}/testexecutions

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output

```
[
   {
      "id":13602,
      "key":"CALC-52",
      "summary":"Test Execution for all tests",
      "self":"http://JIRASERVER/testJira/rest/api/2/issue/CALC-52"
   },
   {
      "id":13600,
      "key":"CALC-53",
      "summary":"Test Execution of v2.0",
      "self":"http://JIRASERVER/testJira/rest/api/2/issue/CALC-53"
   }
]
```

400 BAD_REQUEST : **text/plain** : Returns the error.

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test steps.

Exporting Test Plans

To export the test plans of a Test, you need to specify the key of the Test you wish to export the test plans from

Returns a json with the exported test plans.

Request

PATH PARAMETERS

parameter	type	description
key	*String*	- Test Key

Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/test/{key}/testplans

Responses

200 OK : **text/plain** : Successful. Returns a json.

Example Output

```
[
   {
      "id":13602,
      "key":"CALC-54",
      "summary":"Test Plan for all tests",
      "self":"http://JIRASERVER/testJira/rest/api/2/issue/CALC-54"
   },
   {
      "id":13600,
      "key":"CALC-55",
      "summary":"Test Plan of v2.0",
      "self":"http://JIRASERVER/testJira/rest/api/2/issue/CALC-55"
   }
]
```

400 BAD_REQUEST : **text/plain** : Returns the error.

401 UNAUTHORIZED : **text/plain** : The Xray for JIRA license is not valid.

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred getting the test steps.

To export the test sets of a Test, you need to specify the key of the Test you wish to export the test sets from.