

Test Runs - REST

Test Runs

- [Test Runs](#)
 - [Test Run](#)
 - [Status](#)
 - [Execution Defects](#)
 - [Execution Evidence](#)
 - [Comment](#)
 - [Examples](#)
 - [Test Steps](#)
 - [Assignee](#)

Test Run

- To view a Test Run, you need to specify the ID of the Test Run. Another way to view a Test Run is to specify the Test Exec and Test Issue to which the Test Run belongs.
- When viewing a Test Run that has iterations, Test Steps are not present.
- To update a Test Run, you need to send a JSON with the Test Run fields you want to update.

Return a json that represents the test run.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- id of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/2
```

Responses

200 OK : **text/plain** : Successful. Return a json that represents the test run.

Example Output

```
{
  "id":371,
  "status":"TODO",
  "testKey":"CALC-12",
  "testExecKey":"CALC-13",
  "assignee":"admin",
  "executedBy":"admin",
  "startedOn": "2016-10-11T17:14:03+01:00",
  "finishedOn": "2016-10-24T14:58:35+01:00",
  "duration": 1115072328,
  "defects":[
    {
      "id":15017,
      "key":"test-114",
      "summary":"summary 1",
      "status":"Open"
    },
    {
      "id":15018,
      "key":"test-115",
      "summary":"summary 2",
      "status":"Open"
    }
  ]
}
```

```

],
"evidences":[
  {
    "id":313,
    "fileName":"stuff.txt",
    "fileSize":"0,0 kB",
    "created":"2016-10-24T15:50:57+01:00",
    "author":"admin",
    "fileURL":"http://lenovo-pc:2990/jira/plugins/servlet/raven/attachment/313/stuff.txt"
  },
  {
    "id":314,
    "fileName":"stuff2.txt",
    "fileSize":"0,0 kB",
    "created":"2016-10-24T15:50:57+01:00",
    "author":"admin",
    "fileURL":"http://lenovo-pc:2990/jira/plugins/servlet/raven/attachment/314/stuff2.txt"
  }
],
"comment":"new comment",
"scenarioOutline":"Given I have entered <input_1> into the calculator\r\nAnd I have entered <input_2> into
the calculator\r\nWhen I press <button>\r\nThen the result should be <output> on the screen\r\n\r\n Examples:
\r\n   | input_1 | input_2 | button | output |\r\n   | 20      | 30      | add    | 50      |\r\n 2   | 5       | add     | 7      |\r\n   | 0       | 40      | add    | 40      |",
"examples":[
  {
    "id":1377,
    "rank":0,
    "values":[
      "input_1",
      "input_2",
      "button",
      "output"
    ],
    "status":"PASS",
    "statusDescription":"The test run has passed"
  },
  {
    "id":1378,
    "rank":1,
    "values":[
      "20",
      "30",
      "add",
      "50"
    ],
    "status":"PASS",
    "statusDescription":"The test run has passed"
  },
  {
    "id":1379,
    "rank":2,
    "values":[
      "2",
      "5",
      "add",
      "7"
    ],
    "status":"PASS",
    "statusDescription":"The test run has passed"
  },
  {
    "id":1380,
    "rank":3,
    "values":[
      "0",
      "40",
      "add",
      "40"
    ],
    "status":"PASS",

```

```
        "statusDescription":"The test run has passed"
    }
  ],
  "environments" : [
    "IOS",
    "Android"
  ]
}
```

400 BAD REQUEST : **text/plain** : Returns the error.

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

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

Return a json that represents the test run.

Request

QUERY PARAMETERS

parameter	type	description
testExecIssueKey	String	- key of the Test Execution.
testIssueKey	String	- key of the Test Issue.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun?testExecIssueKey=TEST-123&testIssueKey=TEST-321
```

Responses

200 OK : **text/plain** : Successful. Return a json that represents the test run.

Example Output

```
{
  "id":371,
  "status":"TODO",
  "testKey":"CALC-12",
  "testExecKey":"CALC-13",
  "assignee":"admin",
  "executedBy":"admin",
  "startedOn":"2016-10-24T15:50:57+01:00",
  "defects":[
    {
      "id":15017,
      "key":"test-114",
      "summary":"summary 1",
      "status":"Open"
    },
    {
      "id":15018,
      "key":"test-115",
      "summary":"summary 2",
      "status":"Open"
    }
  ],
  "evidences":[
    {
      "id":313,
```

```

        "fileName": "stuff.txt",
        "fileSize": "0,0 kB",
        "created": "2016-10-24T15:50:57+01:00",
        "author": "admin",
        "fileURL": "http://lenovo-pc:2990/jira/plugins/servlet/raven/attachment/313/stuff.txt"
    },
    {
        "id": 314,
        "fileName": "stuff2.txt",
        "fileSize": "0,0 kB",
        "created": "2016-10-24T15:50:57+01:00",
        "author": "admin",
        "fileURL": "http://lenovo-pc:2990/jira/plugins/servlet/raven/attachment/314/stuff2.txt"
    }
],
"comment": "new comment",
"scenarioOutline": "Given I have entered <input_1> into the calculator\r\nAnd I have entered <input_2> into the calculator\r\nWhen I press <button>\r\nThen the result should be <output> on the screen\r\n\r\n Examples:\r\n  | input_1 | input_2 | button | output |\r\n  2 | 5 | add | 7 | \r\n  | 0 | 40 | add | 40 |",
"examples": [
    {
        "id": 1377,
        "rank": 0,
        "values": [
            "input_1",
            "input_2",
            "button",
            "output"
        ],
        "status": "PASS",
        "statusDescription": "The test run has passed"
    },
    {
        "id": 1378,
        "rank": 1,
        "values": [
            "20",
            "30",
            "add",
            "50"
        ],
        "status": "PASS",
        "statusDescription": "The test run has passed"
    },
    {
        "id": 1379,
        "rank": 2,
        "values": [
            "2",
            "5",
            "add",
            "7"
        ],
        "status": "PASS",
        "statusDescription": "The test run has passed"
    },
    {
        "id": 1380,
        "rank": 3,
        "values": [
            "0",
            "40",
            "add",
            "40"
        ],
        "status": "PASS",
        "statusDescription": "The test run has passed"
    }
],
"environments" : [

```

```
}
    "IOS",
    "Android"
  ]
}
```

400 BAD REQUEST : **text/plain** : Returns the error.

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

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

Update the test run. The fields that can be updated on the test run are: **status**, **comment**, **assignee**, **defects**, **evidences**, **examples** and **steps**.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- ID of the test run.

Example

Example Input 1

```
{
  "status": "FAIL",
  "comment": "new comment",
  "assignee": "ampr",
  "defects": {
    "add": [
      "test-114",
      "test-115",
      "test-116"
    ]
  },
  "evidences": {
    "add": [{
      "filename": "test1.txt",
      "contentType": "plain/text",
      "data": "(base64 encoding...)"
    }, {
      "filename": "test2.txt",
      "contentType": "plain/text",
      "data": "(base64 encoding...)"
    }],
    "remove": ["254", "443"]
  },
  "examples": [{
    "id": "1379",
    "status": "TODO"
  }]
}
```

Example Input 2

```
{
  "status": "FAIL",
  "comment": "new comment",
  "assignee": "ampr",
  "defects": {
    "add": [
      "test-114",
      "test-115",
      "test-116"
    ]
  },
  "evidences": {
    "add": [{
      "filename": "test1.txt",
      "contentType": "plain/text",
      "data": "(base64 encoding...)"
    }, {
      "filename": "test2.txt",
      "contentType": "plain/text",
      "data": "(base64 encoding...)"
    }],
    "remove": ["254", "443"]
  },
  "steps": [{
    "id": "731",
    "status": "TODO",
    "comment": "the comment 1"
  }, {
    "id": "730",
    "status": "PASS",
    "defects": {
      "add": [
        "test-114",
        "test-115",
        "test-116",
        "appId=a364a9c7-9ac0-3183-9175-353c1331692a&issue=SDP-5"
      ]
    },
    "evidences": {
      "add": [{
        "filename": "test1.txt",
        "contentType": "plain/text",
        "data": "(base64 encoding...)"
      }, {
        "filename": "test2.txt",
        "contentType": "plain/text",
        "data": "(base64 encoding...)"
      }],
      "remove": ["274", "543"]
    }
  ]
}
```

Example Input 3

```
{
  "status": "FAIL",
  "assignee": "ampr",
  "defects": {"add": ["appId=a364a9c7-9ac0-3183-9175-353c1331692a&issue=SDP-5"], "remove": ["DCW-9",
"appId=a364a9c7-9ac0-3183-9175-353c1331692a&issue=SDP-1"]},
  "evidences": [],
  "steps": [
    {
      "id": 43,
      "status": "PASS",
      "comment": "",
      "defects": {"add": [], "remove": ["DCW-9"]},
      "evidences": []
    },
    {
      "id": 44,
      "status": "FAIL",
      "defects": {},
      "evidences": []
    },
    {
      "id": 45,
      "status": "FAIL",
      "defects": {},
      "evidences": []
    }
  ]
}
```



Example Request

```
curl -H "Content-Type: application/json" -X PUT -u admin:admin --data @example.json http://yourserver / rest/raven/1.0/api/testrun/1
```

Responses

200 OK : **text/plain** : Successful.

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

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

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

Status

- To view the status of a Test Run, you need to specify the ID of the Test Run.
- To update the status, an additional parameter is required: the Status that you want the Test Run to have.

Return a json that contains the test run status.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- ID of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/status
```

Responses

200 OK : **text/plain** : Successful. Returns the name of the test run Status.

Example Output

PASS

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when getting the test run status.

Updates the test run status.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- ID of the test run.

QUERY PARAMETERS

parameter	type	description
status	String	- The status you want the test run to have.



Example Request

```
curl -H "Content-Type: application/json" -X PUT -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/status?status=PASS
```

Responses

200 OK : **text/plain** : Successful.

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when updating the test run status.

Execution Defects

- To view the defects of a Test Run, you need to specify the ID of the Test Run.
- To add a new defect to the Test Run, a JSON with the defect issue key(s) is required.
- To delete one or more defects from the Test Run, you need to specify the key of the issue you want to delete.

Return a json that contains an array with all the defects the test run has.

Request

PATH PARAMETERS

parameter	type	description
id	<i>integer</i>	- id of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/defect
```

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run defects.

Example Output

```
[
  {
    "id":15017,
    "key":"TEST-114",
    "summary":"summary 1",
    "status":"Open"
  },
  {
    "id":15018,
    "key":"TEST-115",
    "summary":"summary 2",
    "status":"Open"
  },
  {
    "id":15019,
    "key":"TEST-116",
    "summary":"summary 3",
    "status":"Open"
  }
]
```

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when getting the test run defects.

Add new defects to the Test Run.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- id of the test run.

Example

Example Input

```
[  
  "TEST-114",  
  "TEST-115",  
  "TEST-116",  
  "TEST-117",  
  "appId=a364a9c7-9ac0-3183-9175-353c1331692a&issue=SDP-1" (in case of defects from remote JIRA servers)  
]
```



Example Request

```
curl -H "Content-Type: application/json" -X POST -u admin:admin --data @example.json http://yourserver / rest/raven/1.0/api/testrun/2/defect
```

Responses

200 OK : **text/plain** : Successful.

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when inserting the defects.

Remove one or more defects from the test run.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- ID of the test run.
issueIdOrKey	String or Integer	- ID or key of the issue you want to remove from the defects.



Example Request

```
curl -H "Content-Type: application/json" -X DELETE -u admin:admin http://yourserver / rest/raven/1.0/api/testrun/2/defect/TEST-421
```

Responses

200 OK : **text/plain** : Successful.

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when inserting the defects.

Execution Evidence

- To view the execution evidence of a Test Run, you need to specify the ID of the Test Run.
- To add new execution evidence to the Test Run, a JSON is required.
- To delete one or more execution evidence from the Test Run, you need to send a JSON with the file name of the evidence you want to delete. Another way to delete an execution evidence is to specify the evidence ID.

Return a json that contains an array with all the execution evidence the test run has.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- id of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/attachment
```

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run execution evidence.

Example Output

```
[
  {
    "id":253,
    "fileName":"TestdataPoorPerformance.csv",
    "fileSize":"35 kB",
    "created":"2016-10-24T15:50:57+01:00",
    "author":"admin",
    "fileURL": "http://YOURJIRA/jira/plugins/servlet/raven/attachment/253/TestdataPoorPerformance.csv"
  },
  {
    "id":254,
    "fileName":"testfile.txt",
    "fileSize":"35 kB",
    "created":"2016-10-24T16:50:57+01:00",
    "author":"admin"
    "fileURL": "http://YOURJIRA/jira/plugins/servlet/raven/attachment/254/testfile.txt"
  }
]
```

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when getting the evidence.

Add a new evidence to the test run.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- id of the test run.

Example

Example Input

```
{
  "data": "iVBORw0KGgoAAAANSUhEUgAABkIAAA09CAYAAADezXv6AAAAAXNSR0IArs4c6QAA(...base64 file encoding)",
  "filename": "image21.jpg",
  "contentType": "image/jpeg"
}
```



Example Request

```
curl -H "Content-Type: application/json" -X POST -u admin:admin --data @example.json http://yourserver / rest/raven/1.0/api/testrun/2/attachment
```

Responses

200 OK : **text/plain** : Successful.

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when inserting the evidences.

Remove all evidences with the same filename from the test run.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- ID of the test run.

Example

Example Input

image.png



Example Request

```
curl -H "Content-Type: application/json" -X DELETE -u admin:admin --data @example.json http://yourserver / rest/raven/1.0/api/testrun/2/attachment
```

Responses

200 OK : **text/plain** : Successful.

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when deleting the evidences .

Remove the evidence with the given attachment id.

Request

PATH PARAMETERS

parameter	type	description
id	Integer	- id of the test run.
attachmentid	Integer	- id of the attachment we want to delete.



Example Request

```
curl -H "Content-Type: application/json" -X DELETE -u admin:admin http://yourserver / rest/raven/1.0/api/testrun/2/attachment/1
```

Responses

200 OK : **text/plain** : Successful.

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

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when deleting the evidences.

Comment

- To view the comment of a Test Run, you need to specify the ID of the Test Run.

- To update the comment, you need to send a JSON with the comment that you want the Test Run to have.

Return a json that contains the test run comment in raw and rendered state.

Request

PATH PARAMETERS

parameter	type	description
id	<i>integer</i>	- id of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/comment
```

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run comment.

Example Output

```
{
  "raw": "this is the test run comment.",
  "rendered": "<p>this is the test run comment.</p>"
}
```

400 BAD REQUEST : **text/plain** : Returns the error.

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

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

Update the comment and the return a json that contains the updated test run comment in raw and rendered state.

Request

PATH PARAMETERS

parameter	type	description
id	integer	- id of the test run.

Example

Example input

This is the new test run comment.



Example Request

```
curl -H "Content-Type: application/json" -X PUT -u admin:admin --data @example.json http://yourserver / rest/raven/1.0/api/testrun/2 /comment
```

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run comment.

Example Output

```
{
  "raw": "this is the new test run comment.",
  "rendered": "<p>this is the new test run comment.</p>"
}
```

400 BAD REQUEST : **text/plain** : Returns the error.

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

500 INTERNAL SERVER ERROR : **text/plain** : An internal error occurred when updating the test run comment.

Examples

- To view all examples of a Test Run, you need to specify the ID of the Test Run.
- To view or edit a specific test example, check out [Test Examples - REST](#)

Return a json that contains all of the test run examples .

Request

PATH PARAMETERS

parameter	type	description
id	integer	- id of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/example
```

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run examples.

Example Output

```
[
  {
    "id": 5068,
    "rank": 1,
    "values": [
      "0",
      "factorial",
      "1"
    ],
    "status": "PASS",
    "duration": "0 millisec",
    "backgrounds": [
      {
        "id": 1337,
        "rank": 1,
        "type": "background",
        "keyword": "Given ",
        "name": "a calculator I just turned on",
        "duration": "0 millisec",
        "status": "PASS"
      }
    ],
    "hooks": [
      {
        "id": 1338,
        "rank": 2,
        "type": "hook",
        "keyword": "Before",
        "name": "FactorialCalculator.setUp()",
        "duration": "0 millisec",
        "status": "PASS"
      },
      {
        "id": 1339,
        "rank": 3,
        "type": "hook",
        "keyword": "After",
        "name": "FactorialCalculator.tearDown()",
        "duration": "0 millisec",
        "status": "PASS"
      }
    ],
    "steps": [
      {
        "id": 1340,
        "rank": 4,
        "type": "scenario",
        "keyword": "Given ",
        "name": "I have entered 3 into the calculator",
        "duration": "0 millisec",
        "status": "PASS"
      },
      {
        "id": 1341,
        "rank": 5,
        "type": "scenario",
        "keyword": "When ",
        "name": "I press factorial",
        "duration": "0 millisec",
        "status": "PASS"
      }
    ]
  }
]
```



```

        "id": 1342,
        "rank": 6,
        "type": "scenario",
        "keyword": "Then ",
        "name": "the result should be 6 on the screen",
        "duration": "0 millisec",
        "status": "PASS"
    }
]
},
{
    "id": 5069,
    "rank": 2,
    "values": [
        "1",
        "factorial",
        "1"
    ],
    "status": "FAIL",
    "duration": "2 millisec",
    "backgrounds": [
        {
            "id": 1343,
            "rank": 7,
            "type": "background",
            "keyword": "Given ",
            "name": "a calculator I just turned on",
            "duration": "0 millisec",
            "status": "PASS"
        }
    ],
    "hooks": [
        {
            "id": 1344,
            "rank": 8,
            "type": "hook",
            "keyword": "Before",
            "name": "FactorialCalculator.setUp()",
            "duration": "0 millisec",
            "status": "PASS"
        },
        {
            "id": 1345,
            "rank": 9,
            "type": "hook",
            "keyword": "After",
            "name": "FactorialCalculator.tearDown()",
            "duration": "0 millisec",
            "status": "PASS"
        }
    ],
    "steps": [
        {
            "id": 1346,
            "rank": 10,
            "type": "scenario",
            "keyword": "Given ",
            "name": "I have entered 4 into the calculator",
            "duration": "0 millisec",
            "status": "PASS"
        },
        {
            "id": 1347,
            "rank": 11,
            "type": "scenario",
            "keyword": "When ",
            "name": "I press factorial",
            "duration": "0 millisec",
            "status": "PASS"
        }
    ],
    {

```

```

        "id": 1348,
        "rank": 12,
        "type": "scenario",
        "keyword": "Then ",
        "name": "the result should be 65 on the screen",
        "duration": "2 millisec",
        "logError": "java.lang.AssertionError: \nExpected: is <65L>\n      but: was <24L>\r\n\tat org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)\r\n\tat org.junit.Assert.assertThat(Assert.java:956)\r\n\tat org.junit.Assert.assertThat(Assert.java:923)\r\n\tat com.xpandit.automation.cucumber.FactorialCalculator.theResultShouldBeOutputOnTheScreen(FactorialCalculator.java:47)\r\n\tat .Then the result should be 65 on the screen(com/xpandit/automation/cucumber/2_factorial_error.feature:11)",
        "status": "FAIL"
      }
    ]
  }
}

```

400 BAD REQUEST : **text/plain** : Returns the error.

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

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

Test Steps

- To view all the test steps of a Test Run, you need to specify the ID of the Test Run.
- To view or edit a specific test step, check out [Test Steps - REST](#)

Return a json that contains all of the test run steps.

Request

PATH PARAMETERS

parameter	type	description
id	<i>integer</i>	- id of the test run.



Example Request

```
curl -H "Content-Type: application/json" -X GET -u admin:admin http://yourserver/rest/raven/1.0/api/testrun/3/step
```

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run steps.

Example Output

```

[
  {
    "id": 729,
    "index": 1,
    "step": {
      "raw": "ssaf",
      "rendered": "<p>ssaf</p>"
    },
    "data": {
      "raw": "asdf",
      "rendered": "<p>asdf</p>"
    },
    "result": {
      "raw": "asdfasdf",
      "rendered": "<p>asdfasdf</p>"
    }
  }
]

```

```

    },
    "attachments": [
    ],
    "status": "EXECUTING",
    "comment": {
        "raw": "asdasdasdasdasd",
        "rendered": "<p>asdasdasdasdasd</p>"
    },
    "defects": [
        {
            "id": 15018,
            "key": "CALC-115",
            "summary": "Ad-hoc execution for CALC-93",
            "status": "Open"
        }
    ],
    "evidences": [
        {
            "id": 216,
            "fileName": "1 (24).feature",
            "fileSize": "0,4 kB",
            "created": "2016-10-24T15:50:57+01:00",
            "author": "admin"
            "fileURL": "http://YOURJIRA/jira/plugins/servlet/raven/attachment/253/1 (24).feature"
        }
    ],
    "actualResult": {
        "raw": "Actual result on step 1",
        "rendered": "<p>Actual result on step 1</p>"
    }
},
{
    "id": 730,
    "index": 2,
    "step": {
        "raw": "asfd",
        "rendered": "<p>asfd</p>"
    },
    "data": {
        "raw": "aaa",
        "rendered": "<p>aaa</p>"
    },
    "result": {
        "raw": "aaaa",
        "rendered": "<p>aaaa</p>"
    },
    "attachments": [
    ],
    "status": "FAIL",
    "comment": {
        "raw": "aaaaa\\n",
        "rendered": "<p>aaaaa</p>"
    },
    "defects": [
    ],
    "evidences": [
    ],
    "actualResult": {
        "rendered": ""
    }
}
]

```

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

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

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

Assignee

- To view the assignee of a Test Run, you need to specify the ID of the Test Run.
- To update the Test Run assignee, an additional parameter is required: the key of the user that you want the Test Run to be assigned to.

Return a json containing the test run assignee.

Request

PATH PARAMETERS

parameter	type	description
id	<i>integer</i>	- id of the test run.



Example Request

curl -H "Content-Type: application/json" -X GET -u admin:admin <http://yourserver/rest/raven/1.0/api/testrun/3/assignee>

Responses

200 OK : **text/plain** : Successful. Returns a json with the test run assignee.

Example Output

```
{
  "assigned": true,
  "assignee": "ampr"
}
```

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

404 NOT_FOUND : **text/plain** : Test run not found.

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when getting the test run assignee.

Updates the test run assignee.

Request

PATH PARAMETERS

parameter	type	description
id	<i>integer</i>	- ID of the test run.

QUERY PARAMETERS

parameter	type	description
user	<i>String</i>	- User's key or username from the user to be assigned to the test run Xray will first try to find the user by its key and if it don't then it will try to find the user by its username



Example Request

curl -H "Content-Type: application/json" -X PUT -u admin:admin <http://yourserver/rest/raven/1.0/api/testrun/3/assignee?user=ampr>

Responses

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

Example Output

```
{
  "assigned": true,
  "assignee": "ampr"
}
```

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

404 NOT_FOUND : **text/plain** : Test run not found.

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

500 INTERNAL_SERVER_ERROR : **text/plain** : An internal error occurred when updating the test run assignee.