

How to manage attachments via REST API



REST API methods are not available in Smart Attachments 2.X+ versions. Please use [Java API](#) instead.

- Subscribe to receive notifications about changes in REST API of Smart Attachments
- [Fetching categories with attachments within the issue](#)
- [Moving attachments from one category to another](#)
- [Sending attachments to email](#)
- [Renaming Attachments](#)
- [Deleting attachments from the category of the same issue](#)
- [Deleting document revisions](#)
- [Archiving attachments](#)
- [Putting attachments to categories before upload](#)
- [Getting attachment metadata](#)
 - [AttachmentMetadata](#)
 - [Metadata](#)
 - [FieldName](#)
 - [StatusCode](#)
- [Getting metadata for all attachments from the issue](#)
- [Updating metadata for attachments](#)

Subscribe to receive notifications about change
Smart Attachments

* indicates required

Email Address *

Full Name *

Company

Fetching categories with attachments within the issue



If you try to use this old cURL request with the newer version of Smart Attachments 1.17.2 (and newer), you will receive the following error:

HTTP 400 Bad request

This method is deprecated. Please, use 'GET /rest/attach-cat/1.0/attachments'

You can get the list of categories with attachments for the specific JIRA issue.

The pattern of request looks like this:

```
curl -D- -u username:password --request GET JIRA.baseUrl/rest/attach-cat/1.0/attachments/get?issueKey=issue_key
```

In this request you need to enter the correct value for the following parameter:

Parameter	Description
issueKey	Key of the JIRA issue which you want to see the list of categories with attachments.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request GET "https://yourjira.com/rest/attach-cat/1.0/attachments/get?issueKey=SMART-97"
```

As a response, you will receive the following JSON string:

```
{
  "categories": [
    {
      "id": 1,
      "name": "Documents",
      "attachments": {
        "ids": [10100, 10009, 10007, 10004, 10003, 10002],
        ...
      },
      hidden: null,
      groups: ["group1", "group2", ...],
      roles: ["project-role1", ...],
      users: ["user1", ...],
      ...
    },
    {
      "id": 2,
      "name": "Logs",
      "attachments": {
        "ids": null,
        ...
      },
      hidden: true,
      groups: [],
      roles: ["developers", ...],
      users: [],
      ...
    },
    ...
  ]
}
```

Flags contained in the response indicate the following:

- **hidden** - indicates that the category has access restrictions. The following values are possible:
 - **true** - indicates that the category is hidden from the current user due to the set restrictions.
 - **null** - indicates that the category has no access restrictions specified.
- **groups, roles, users** - groups, roles and users who can view the category. Nothing is displayed for the categories having no access restrictions set.

You can get the list of categories with attachments for the specific JIRA issue.

The pattern of request looks like this:

```
curl -D- -u username:password --request GET JIRA.baseUrl/rest/attach-cat/1.0/attachments?issueKey=issue_key
```

In this request you need to enter the correct value for the following parameter:

Parameter	Description
issueKey	Key of the JIRA issue which you want to see the list of categories with attachments.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request GET "https://yourjira.com/rest/attach-cat/1.0/attachments?issueKey=SMART-97"
```

As a response, you will receive the following JSON string:

```
{
  "categories":[
    {
      "id":1,
      "name":"Sample",
      "position":0,
      "attachments":[
        10000,
        10002
      ],
      "documents":[
        {
          "id":1,
          "name":"document",
          "attachmentIds":[
            10003,
            10004,
            10005
          ]
        }
      ]
    },
    {
      "id":2,
      "name":"Admin Cat",
      "position":1,
      "attachments":[
        10010,
        10011
      ],
      "documents":[]
    },
    {
      "id":0,
      "name":"Uncategorized",
      "position":5,
      "attachments":[]
    }
  ]
}
```

Moving attachments from one category to another

You can move one or multiple attachments from one to another category of the same issue.

The pattern of request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseUrl/rest/attach-cat/1.0/attachments/move" --data
"issueKey=issue_key&toCatId=category_id&attachments=attachment_id_1&attachments=attachment_id_2"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issueKey	Key of the JIRA issue for moving attachments.

toCatId	ID of the category which attachments are moved to.
attachments	ID's of attachments you want to move.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/move" --data "issueKey=SMART-97&toCatId=0&attachments=10801&attachments=10802"
```

No response is returned.

Sending attachments to email

You can send one or multiple attachments to one or multiple email recipients. The add-on API allows you to optionally specify email address using CC and BCC fields.

The pattern of request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseURL/rest/attach-cat/1.0/attachments/send/mail" --data "issueKey=issue_key&subject=YourSubject&message=YourMessage&ids=attachment_id&to=example@email.com&cc=cc@email.com&bcc=bcc@email.com&sendWithoutAttachments=false"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issueKey	Key of the JIRA issue which attachments you want to send to email.
subject	Email subject.
message	Email message.
ids	ID's of attachments you want to email.
to	Email address of the person who you send attachments to. Use ', ' (comma with space) to enter multiple emails.
cc	CC (carbon copy) email address which you want to send the attachments to. Use ', ' (comma with space) to enter multiple emails.
bcc	BCC (blind carbon copy) email address which you want to send the attachments to. Use ', ' (comma with space) to enter multiple emails.
sendWithoutAttachments	Allows you to send an email without attachments when there are no attachments with the specified ID's. It may accept two values:

Here you can see an example of the request:

```
curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/send/mail" --data "issueKey=SMART-87&subject=Documents for review&message=Please take a look at attached documents and let me know your opinion.&ids=10800&ids=10801&ids=10802&to=pjacobs@gmail.com&cc=astone@yahoo.com&bcc=mbrown@gmail.co&sendWithoutAttachments=false"
//with one email address in each parameter

curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/send/mail" --data "issueKey=SMART-87&subject=Documents for review&message=Please take a look at attached documents and let me know your opinion.&ids=10800&ids=10801&ids=10802&to=pjacobs@gmail.com, lbrown@yahoo.com, mwhite@gmail.com&sendWithoutAttachments=true"
//with multiple emails in one parameter
```

No response is returned.

Renaming Attachments

You can rename the attachments in JIRA issues.

The pattern of request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseURL/rest/attach-cat/1.0/attachments/rename" --data "issueKey=issue_key&id=attachmentID&catId=CategoryID&name=newName"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issueKey	Key of the JIRA issue which attachments you want to rename.
id	ID of the attachment you want to rename. Omit this parameter if you want to rename a specific document but not a specific attachment.
catId	ID of the category where the attachment for renaming is stored.
name	A new name for the attachment.
	Use it instead of 'id' when you want to rename the document instead of a specific attachment.

Here you can see an example of the request:

```
curl -D- -u username:password --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/rename" --data "issueKey=SMART-87&id=10805&catId=2&name=Financial Report.docx"
```

No response is returned.

Deleting attachments from the category of the same issue

You can delete one or multiple attachments from the issue.

The pattern of request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseurl/rest/attach-cat/1.0/attachments/remove" --data "issueKey=issue_key&ids=attachment_id_1&ids=attachment_id_2"
```

In this request you need to enter the correct value for the following parameter:

Parameter	Description
ids	ID's of attachments you want to delete.
issueKey	Key of the JIRA issue which attachments you want to delete from.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/remove" --data "issueKey=SMART-87&ids=161105&ids=161107"
```

No response is returned.

Deleting document revisions



Please note that the whole document with all its revisions will be removed when you specify ID of any revision of this document.

You can delete one or multiple document revisions from the issue.

The pattern of request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseurl/rest/attach-cat/1.0/attachments/remove/document" --data "issueKey=issue_key&id=attachment_id_1&id=attachment_id_2"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issueKey	Key of the JIRA issue which you want to delete document revisions from.
id	ID of document/attachment you want to delete with all its revisions.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/remove/document" --data "issueKey=SMART-87&id=10801&id=10802"
```

No response is returned.

Archiving attachments

You can archive one or multiple attachments stored within your issue.

The pattern of the request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseurl/rest/attach-cat/1.0/attachments/archive" --data "issueKey=issue_key&ids=attachment_id&cat=category_id&name=archive_name"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issueKey	Key of the JIRA issue which you want to see the list of categories with attachments.
ids	ID's of attachments and document revisions you want to archive.
docIds	ID's of documents you want to archive.
cat	ID of the category where you want to place the archive with attachments.
name	Name of the archive. The ZIP extension will be automatically appended to the archive name.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments/archive" --data "issueKey=SMART-87&ids=11508&ids=11510&cat=2&name=file-backup"
```

Putting attachments to categories before upload



Attachments will be placed to the specific attachment category only when the user who has run this request and then uploaded attachments was the same user.

You can map attachments to the specific category by their file name before files are actually uploaded to the issue.

The pattern of the request looks like this:

```
curl -D- -u username:password --request POST "JIRA.baseurl/rest/attach-cat/1.0/attachments" --data "issueKey=issue_key&issueId=issue_Id&tempAttachments=yourJSON_string"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issueKey	Key of the JIRA issue.
issueId	ID of the JIRA issue (it's optional)
tempAttachments	The names of the attachments in JSON format "[{"name":"your attachment name","categoryId": "your category ID"}]" that should be placed to specific categories once uploaded.

Here you can see an example of the request:

```
//Use the \" ... \" construction to escape file names and categories in the JSON string (character \" (double quote) replace by \" in data param)
curl -D- -u admin:admin --request POST "https://yourjira.com/rest/attach-cat/1.0/attachments" --data "issueKey=TAP-1&issueId=10000&tempAttachments=[{\"name\":\"av-29040.png\", \"categoryId\":0},{\"name\":\"document1.docx\", \"categoryId\":3}]"
```

Getting attachment metadata

You can get metadata for specific attachments.

The pattern of the request looks like this:

```
curl -D- -u username:password --request GET "JIRA.baseurl/rest/attach-cat/1.0/metadata/" --data "attachmentId=attachment_id"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
attachmentId	ID of the attachment which you want to get metadata for.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request GET "https://yourjira.com/rest/attach-cat/1.0/metadata/" --data "attachmentId=10406"
```

As a response, you will get a JSON string:

```
{
  "attachmentId": 10406,
  "values": [
    {
      "name": "documentId",
      "value": "SIFE-10PGR-2021"
    },
    {
      "name": "description",
      "value": "10-pager for LinkedIn Ads"
    },
    {
      "name": "revision",
      "value": "1.1"
    },
    {
      "name": "status",
      "value": "5"
    },
    {
      "name": "signedBy",
      "value": "Larry Stenfer"
    }
  ]
}
```

AttachmentMetadata

Name	Type
attachmentId	long
values	JSON string

Metadata

Name	Type
name	FieldName
value	String(255)

FieldName

Supported values for FieldName

Name	Allowed value
documentId	String(255)
decription	String(255)
revision	String(255)
status	StatusCode
link	String(255)
signedBy	String(255)

StatusCode

Status	Code
Draft	1

In Progress	2
In Review	3
Rejected	4
Approved	5
Published	6
Signed	7

Getting metadata for all attachments from the issue

You can get metadata for all attachments within a specific issue.

The pattern of the request looks like this:

```
curl -D- -u username:password --request GET "JIRA.baseurl/rest/attach-cat/1.0/metadata/issue/" --data "issuesId=issue_id"
```

In this request you need to specify the correct values for the following parameters:

Parameter	Description
issuesId	ID of the attachment which you want to get metadata for.

Here you can see an example of the request:

```
curl -D- -u admin:admin --request GET "https://yourjira.com/rest/attach-cat/1.0/metadata/issue/" --data "issuesId=10401"
```

As a response, you will get a JSON string:

```

{
  "issueId": 10401,
  "values": {
    "10406": {
      "attachmentId": 10406,
      "values": [
        {
          "name": "documentId",
          "value": "SIFE-10PGR-2021"
        },
        {
          "name": "description",
          "value": "10-pager for LinkedIn Ads"
        },
        {
          "name": "revision",
          "value": "1.1"
        },
        {
          "name": "status",
          "value": "5"
        },
        {
          "name": "signedBy",
          "value": "Larry Stenfer"
        }
      ]
    },
    "10408": {
      "attachmentId": 10408,
      "values": [
        {
          "name": "documentId",
          "value": "SIFE-MKT-PRSNT-2021"
        },
        {
          "name": "description",
          "value": "Presentation of capabilities of Spreadsheet Issue Field Editor"
        },
        {
          "name": "revision",
          "value": "1.0"
        },
        {
          "name": "status",
          "value": "6"
        },
        {
          "name": "signedBy",
          "value": "Vadim Rutkevich"
        }
      ]
    }
  }
}

```

Updating metadata for attachments

You can update metadata of a specific attachment. To remove the entire metadata for an attachment, just submit a request with empty values.

The pattern of the request looks like this:

```

curl -D- -u username:password --request POST "JIRA.baseurl/rest/attach-cat/1.0/metadata/" --data
"issuesId=issue_id"

```

Request headers

Header name	Value
Content-Type	application/json

Request payload should be a JSON string, as follows:

```
{ "attachmentId": 10400, "values": [ { "name": "documentId", "value": "SIFE-YTB-BNR-2021" }, { "name": "description", "value": "Banner for Youtube video about Spreadsheet Issue Field Editor" }, { "name": "revision", "value": "1.0" }, { "name": "signedBy", "value": "Vadim Rutkevich" }, { "name": "status", "value": "6" } ] }
```