

/api/{version}/File

There is a one initial request needed and then with additional requests you can upload parts of a file. The size of a part (except the last one) is 4194304 bytes. This API should be used only for large files (100MB+). Uploaded file will be in our system available for 1 day.

All requests must contain headers with valid values:

1.X-VITR-ACCOUNT-TOKEN

2.X-VITR-SESSION-TOKEN

Initial request

Request type: POST

This request must contain form data which has these fields:

1.**ufname** – contains name of a file (for example "testPDF.pdf")

2.**umime** – contains mime type of file (for example "application/pdf")

3.**usize** – contains size of a file (for example "391211402")

4.**packetcount** – contains the total number of packets (Original file size / chunk file size)

The request content's type is "**multipart/form-data**".

This initial request will return a response:

```
{"Guid":"a6a02a88-5c46-4845-ba92-3cd13563b021","Packet":0}
```

Packet request (this needs to be in loop until it uploads all the chunks)

Request type: POST

We can calculate total packet count by $\text{<fileSize> divided by <packetSize>}$ which is in our system 4194304 bytes. So for file of size 391211402 bytes we get this:

$391211402 / 4194304 = 93.2720665932 \leq 94$ packet requests

This request contains form data with a part of file. The form field name can be anything (for example "file").

We should send packets from 1 to N one by one.

There are needed parameters in the URL:

1.**guid** – id of the file obtained in the initial request (for example "a6a02a88-5c46-4845-ba92-3cd13563b021")

2.**packet** – which part of the file this request contains (starting from 1)

3.**total** – number of total packages which will be sent to the server (for example "94")

**Parse the above parameters in Query Parameters and attach the chunk file in form-data.*

The response will look like this for the first packet sent:

```
{"Guid":"a6a02a88-5c46-4845-ba92-3cd13563b021","Packet":1}
```

The "Packet" field contains a packet which was received by server so we can send "Packet" + 1 next. If the value in "Packet" field will be "-1" there was an error and we have to start again.

If the "Packet" field value equals to total packet count, you successfully uploaded the file.

/api/{version}/Doc

Upload request

Request type: POST

Now you can use the doc upload API to create the content, just instead of "rawfile" in form data add "filename" which contains the <guid>.<extension> string (for example "a6a02a88-5c46-4845-ba92-3cd13563b021.pdf").

Reference the json example below:

```
{"filename": "a6a02a88-5c46-4845-ba92-3cd13563b021.pdf", "settings":  
{"Id":"","FolderId":"","PolicyId":"","FileName":"ORIGINALFILENAME","Title":" ORIGINALFILENAME",  
"SkipPages":"","ExternalKey":"","VersionName":"Version: 1","EnableVersionNotification":true}
```

**Both filename and settings field are required*