



GATE SCANNER Rest API Server Version 3.02 Guide

Your gate to a safe system



Sasa Software Ltd.
Kibbutz Sasa
M.P. Merom Hagalil 13870
Israel

Telephone: +972-4-6918959
Fax: +972-4-6918876
Email: info@sasa-software.com
Web: www.sasa-software.com

The information contained in this document, or any addendum or revision thereof is proprietary of Sasa Software Ltd. and is subject to all relevant copyright, patent and other laws and treaties protecting intellectual property, as well as any specific agreement protecting Sasa Software Ltd. rights in the aforesaid information. Any use of this document or the information contained herein for any purposes other than those for which it was disclosed is strictly forbidden.

Sasa Software Ltd. reserves the right, without prior notice or liability, to make changes in equipment design or specifications.

Sasa Software Ltd. assumes no responsibility for the use thereof nor for the rights of third parties, which may be affected in any way by the use thereof.

This document may contain flaws, omissions or typesetting errors; no warranty is granted nor liability assumed in relation thereto unless specifically undertaken in Sasa Software Ltd.'s sales contract or order confirmation.

Information contained herein is periodically updated and changes will be incorporated into subsequent editions. If you have encountered an error, please notify Sasa Software Ltd.

All specifications are subject to change without prior notice.

© Copyright by Sasa Software Ltd., 2020. All rights reserved worldwide.

Document History

Version Number	Version Date	Author	Description
1	May, 2017	Elad Leshem	Version: 1.0.0
2	July, 2018	Elad Leshem	Version: 2.1.0
3	December, 2018	Elad Leshem	Version: 2.4.0
4	December, 2019	Elad Leshem	Version: 3.0.0
5	May, 2020	Yevgeny Levin, Elad Leshem	Version: 3.02.0

Table of Contents

Gate Scanner® Rest API Documentation	9
Introduction	9
What is Gate Scanner REST API Service	9
Gate Scanner Rest API URIs (Base Address: https://<BaseAddress>/Scanner.svc/)	11
Actions Detail	11
Introduction	11
Upload File	12
Upload File action.....	12
1. Upload	13
POST v2/upload/file	13
Request Information.....	13
URL Example.....	13

URI Parameters.....	13
Body Parameters	13
2. Base64 Upload	13
POST v3/upload/file/base64.....	13
Request Information.....	13
URL Example.....	13
URI Parameters.....	13
Body Parameters	14
Request Formats for both options (General Upload and Base64 Upload).....	14
Response Information	16
Resource Description	16
Response Formats	16
Get Scan Info	17
GET v3/scan/{attempt}/{jobID}.....	17
Request Information.....	17
URL Example.....	17
URI Parameters.....	17
Body Parameters	17
Response Information	18
Resource Description	18
Response Formats	18
Download File	21
1. General Download	21
GET v3/download/file/{jobID}	21
Request Information.....	21
URL Example.....	21
2. Base64 Download.....	21

GET v3/download/file/base64/{jobID}.....	21
Request Information.....	21
URL Example.....	21
URI Parameters.....	22
Body Parameters.....	22
Request Formats.....	22
Response Information.....	23
Resource Description.....	23
Response Formats.....	23
Clean Server Data.....	24
DELETE v3/delete/{jobID}.....	24
Request Information: This action will delete server temporary data if exist, it will not delete the job file or logs in engine.	24
URL Example.....	24
URI Parameters.....	24
Body Parameters.....	24
Response Information.....	24
Resource Description.....	24
Response Formats.....	25
General Resource Description.....	26
JobInfo.....	26
ScansLog.....	26
ScanStatus.....	29
ScanFileResult.....	30
Scenarios Samples.....	31
Introduction.....	31
Single non-chunk file process (Upload, Job Status, Download, Delete).....	31

Batch Upload, scan and download 2 files	32
Upload file 1: testDoc.docx	32
Request URL	32
Request Parameters	32
Body Parameters	32
Request Header.....	33
Server Response	33
Upload file 2: txtTest.txt	34
Request URL	34
Request Parameters	34
Body Parameters	34
Request Header.....	34
Server Response	35
Scan to end, iteration 1	35
Request URL	36
Request Parameters	36
Request Header.....	36
Server Response	36
Scan to end, iteration 2.....	37
Request URL	37
Request Parameters	37
Request Header.....	37
Server Response	37
Download file 1: testDoc.docx	40
Request URL	40
Request Parameters	40
Request Header.....	40

Server Response	40
Download file 2: txtTest.txt	41
Request URL	41
Request Parameters	41
Request Header	41
Server Response	41
Batch Upload with chunks of a big file (one out of two files), Scan and Download Scenario.	42
Upload file 1: DocMParts.docx	42
Part 1	42
Request URL	42
Request Parameters	42
Body Parameters	42
Request Header	42
Server Response	43
Part 2	43
Request URL	43
Request Parameters	43
Body Parameters	43
Request Header	44
Server Response	44
Part 3	45
Request URL	45
Request Parameters	45
Body Parameters	45
Request Header	45
Server Response	45
Upload file 2: ScanTest.txt	46

Request URL	46
Request Parameters	46
Body Parameters	46
Request Header	46
Server Response	47
Scan to end, iteration 1	47
Request URL	47
Request Parameters	48
Request Header	48
Server Response	48
Scan to end, iteration 2	48
Request URL	49
Request Parameters	49
Request Header	49
Server Response	49
Download file 1: DocMParts.docx	52
Request URL	52
Request Parameters	52
Request Header	52
Server Response	52
Download file 2: ScanTest.txt	53
Request URL	53
Request Parameters	53
Request Header	53
Server Response	53

Gate Scanner® Rest API Documentation

Introduction

Thank you for choosing Gate Scanner® REST API (Application Programming Interface) by Sasa Software.

Gate Scanner® treats every incoming file and email as suspicious, performing deep threat scans, transforming files into a safe, neutralized and harmless copy, that ensures security (D-CDR = Deep Content Disarm and Reconstruction).

Gate Scanner® prevents unknown and undetectable malicious code attacks, including ransomware, while maintaining full usability, functionality and visibility of the files.

The Gate Scanner® REST API enables you to integrate your applications with our D-CDR technology for Non-Windows and Windows systems.

By using the API you can send files from your application, and receive the secured file based on HTTPS and JSON Content-Type.

This enhances the security of your solutions and your customers, by adding a proven layer of security against advanced file-based threats.

The documentation below describes our REST API to Gate Scanner®'s engines -

Including sending files, receiving scan results, receiving scanned files, using scanning profiles.

Included are several programming client examples described how to call Gate Scanner® Rest API actions.

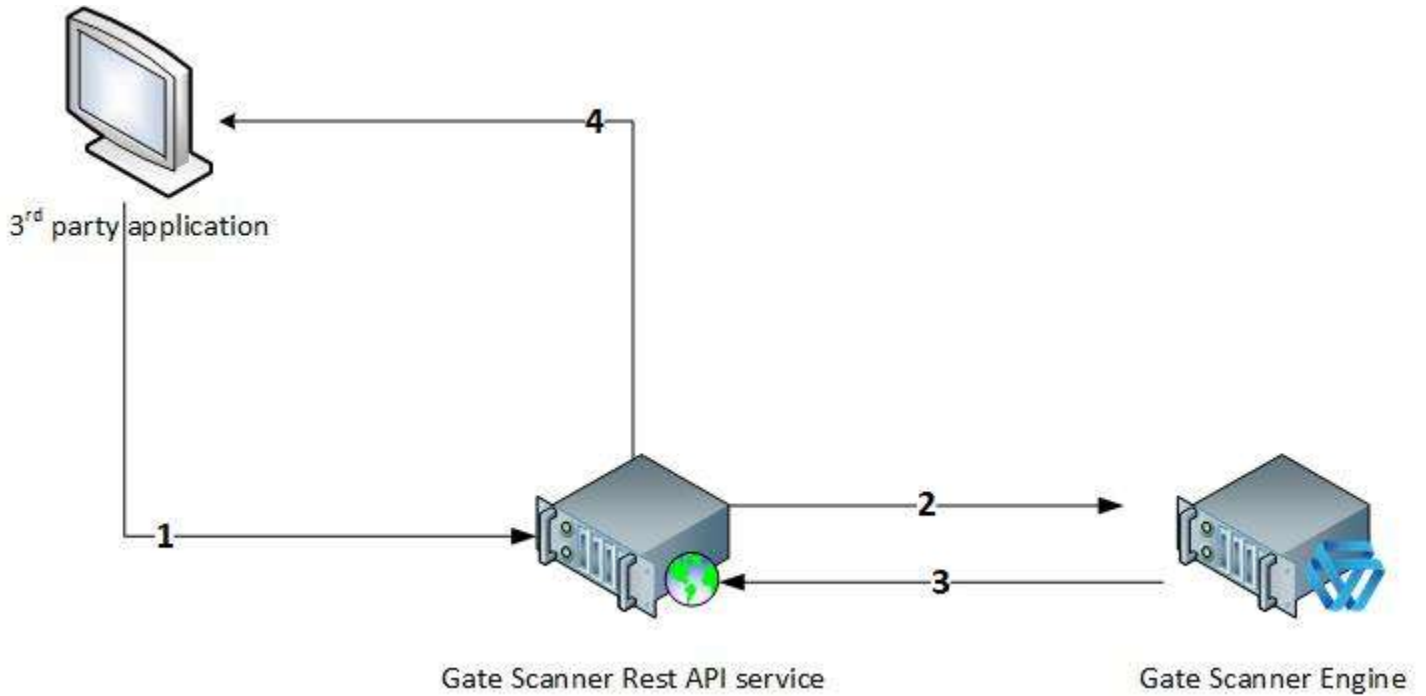
What is Gate Scanner REST API Service

Gate Scanner (GS) Secure API server, also known as Gate Scanner Direct server, allows applications to use Windows Communication Foundation (WCF) for sending file(s) to be scanned and/or converted in the Gate Scanner Engine(s). After completing the scanning process, the clean and converted file(s) are sent back to the application.

One GS WCF server can accept file(s) from multiple sources and multiple applications, using a grid of GS Engines which perform the scanning and conversion. Files are listed and sent to the GS Engines in a batch of files, not a single file at a time. This allows the GS WCF Server to support large amounts of files and capacity.

Using Gate Scanner Rest API service will allow any 3rd party application to use Gate Scanner (GS) Secure API server and not only Windows based applications (WCF is a Windows based service – Windows Communication Foundation)

The diagram below describes the flow of the system:



1. File is submitted for scanning to the Rest API service.
2. Rest API service takes the streamed file(s) and puts it in a folder for the GS Engines service.
3. GS Engine creates a scan job from the file(s) and it completes the scan according to the policy and returns the file(s) and logs to the GS REST API service.
4. File(s) are streamed back to the sender.

Gate Scanner Rest API URIs (Base Address: <https://<BaseAddress>/Scanner.svc/>)

API Actions	Backwards Compatibility	Description
POST v3/upload/file	Version 2 (v2)	Upload file Binary.
POST v3/upload/file/base64	No.	Upload file Base64 Encode.
GET v3/scan/{attempt}/{jobID}	Version 2 (v2)	Check status and logs for the job.
GET v3/download/file/{jobID}	Version 2 (v2)	Download file Binary.
GET v3/download/file/base64/{jobID}	No.	Download file Base64 Encode.
DELETE v3/delete/{jobID}	Version 2 (v2)	Delete data on server.

Actions Detail

Introduction

The Gate Scanner® Rest API Exposed simple Uri's that represents an atomics actions in the Gate Scanner Server.

Bellow a descriptions of these actions list, include the HTTPS Method (POST\GET\DELETE), request information and response information.

Upload File

Upload File action

This action triggers the scanning process.

1. **Big Files** (Upload in chunks) the Gate Scanner® REST API support out of the box upload in chunks, in order to use chunks, the request just need to:
 - Include the **F-Size (full file size)** header property, see description in the Header section bellow.
 - Requests Body with the file parts of the file (in synchronized iteration manner), it can be loaded with different sizes of parts.

The default value of our service for maximum single part size or single full file size is 1GB (Depends also on the web-server max message configuration.), it is **recommended**:

- a. Split the file if it is greater than 10-20MB,
 - b. Load parts that not greater than 5-10MB.
2. Batch Jobs- The GS REST API support scanning batch of files, in order to use this option:
 - Please use the **ScanJob (Batch only)** header property, see description in the Header section bellow.
 - Use **JobID (after first file in batch uploaded)** header property, see description in the Header section bellow.

1. Upload

POST v2/upload/file

Request Information

URL Example

<https://<Base Address>/Scanner.svc/v3/upload/file>

URI Parameters

None.

Body Parameters

Stream\Binary - The File Bytes (without any boundaries, do NOT use Content-Type: multipart).

2. Base64 Upload

POST v3/upload/file/base64

Request Information

URL Example

<https://<BaseAddress>/Scanner.svc/v3/upload/file/base64>

URI Parameters

None.

Body Parameters

Stream\Binary with the file Base64 text.

The Binary without any boundaries, do NOT use content-type multipart.

Request Formats for both options (General Upload and Base64 Upload)

HTTP Header Parameters

Name	Required\Optional	Description	Type
FileName	Required	The filename, limit of 200 chars with suffix\extension.	String
GS-Key	Required	The API Key.	String
AppInfo	Required	The application info will be supplied by the caller application. The application info should be per application name for later statics and scan history analysis.	String
ProfileID	Required	For optimal scanning efficiency, supplied by Sasa-Software. The Gate Scanner engine profile ID.	String
F-Size	Required	The full size of file in bytes. In case of upload in parts (chunks) the REST API will check if the file was fully uploaded by comparison between file size and the parts that already loaded. In case Scan job header property does not exist, the REST API will assume it is a single upload and after the last part loaded, it will scan the job. (See ScanJob parameter in current section)	String
ScanJob	Optional	Needed only in a batch based request (Do not use this header property in case of a single file upload.) Boolean property. If true the REST API will start scanning the job, set false value if more file\s will be sent in the batch .	String (Valid values: true/false)

F-Password	Optional	The REST API Server supports files with password. Use this header property only for files with password.	String
JobID	Optional	Needed only in a batch for 2 nd file and above. The client will get it from the server after first file upload request succeeded.	String
Content-Type		text/plain OR application/x-www-form-urlencoded etc'	String

Sample:

Sample of header.

```
{
  FileName: testImage.jpg
  GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92
  AppInfo: AppInfo
  profileID: 4
  F-Size: 6677
  Accept: application/json
}
```

Response Information

Resource Description

[JobInfo](#)

Response Formats

application/json

Sample:

```
{
  "JobDescription":null,
  "JobError":null,
  "JobID":"9ea2c45d-af7a-4b1d-b015-216788ec5576",
  "PlaceInQueue":-99,
  "ScanProgress":0,
  "ScanStageProgress":0,
  "Scan_Status_c":1,
  "ScansLogArr": null
}
```


Get Scan Info

GET v3/scan/{attempt}/{jobID}

Request Information

URL Example

https://<Base Address>/Scanner.svc/v3/scan/1/05ce1e74-46bd-4f9f-8602-15f9e96090c7

URI Parameters

Name	Description	Type	Additional information
Attempt	Index for each call.	String	Required - Increase the value for each call.
JobID	The server Job ID.	String	Required - The Job ID that was given by the engine in the upload phase.

Body Parameters

None.

Response Information

Resource Description

[JobInfo](#)

Response Formats

application/json

Sample:

```
{ "JobDescription": null,
  "JobError": null,
  "JobID": "a74a65c5-c012-422f-b8a7-0efd1c2ff237",
  "PlaceInQueue": 0,
  "ScanProgress": 0,
  "ScanStageProgress": 0,
  "Scan_Status_c": 3,
  "ScansLogArr":
  [ {
    "Base64ObjData": null,
    "Base64ObjName": null,
    "DestinationFileType": null,
    "DurationSeconds": "0.016",
    "EmbeddedLogs": null,
    "EmbeddedParentName": "",
    "FileHASH": "e12c811bfb4426a55664520f83d81be2de61a31b5b9760201134cb503fd92e21",
    "FileName": "testImage.jpg",
    "FileNewHASH": "e8957d4502a10c7b75e4eac2e6d3cc601b4e7faf55459b33d254e92e81dda2a2",
    "FileNewSizeBytes": "6330",
    "FileSizeBytes": "6677",
    "FullFilePath": "a74a65c5-c012-422f-b8a7-0efd1c2ff237\\testImage.jpg",
```

```
"InnerLogs":null,
"IsReAttached":false,
"LogErrorMessage":null,
"LogMessages":
[
" (C=Image) [TYPE=JPEG;VR=72;HR=72;H=32;W=600;BPP=24bppRgb ] -|- Phase 2 Image -|- Ended OK -> Image Engine - OK",
"Passed Description : <CONVERTED IMG-TO-IMG>\u0009[OK]"
],
"MiddleFormats":["JPG","JPG"],
"OutputFileName":"a74a65c5-c012-422f-b8a7-0efd1c2ff237\\testImage.jpg",
"OutputFullPath":"636661428386281277_9\\a74a65c5-c012-422f-b8a7-0efd1c2ff237\\testImage.jpg",
"ReAttachedToParent":"","
"Result":2,"SourceFileType":"jpg",
"SourceTrueFileType":"jpeg;jpg",
"StartJobDateTime":"\\/Date(1530535244000+0300)\/"
},
{
"Base64ObjData":null,
"Base64ObjName":null,
"DestinationFileType":null,
"DurationSeconds":"0.016",
"EmbeddedLogs":null,
"EmbeddedParentName":"","
"FileHASH":"e12c811bfb4426a55664520f83d81be2de61a31b5b9760201134cb503fd92e21",
"FileName":"testImage.jpg",
"FileNewHASH":"e8957d4502a10c7b75e4eac2e6d3cc601b4e7faf55459b33d254e92e81dda2a2",
"FileNewSizeBytes":"6330",
"FileSizeBytes":"6677",
"FullPath":"a74a65c5-c012-422f-b8a7-0efd1c2ff237\\testImage.jpg",
```

```
"InnerLogs":null,  
"IsReAttached":false,  
"LogErrorMessage":null,  
"LogMessages":[" (C=Image) [TYPE=JPEG;VR=72;HR=72;H=32;W=600;BPP=24bppRgb ] -|- Phase 2 Image -|-  
Ended OK -> Image Engine - OK",  
"Passed Description : <CONVERTED IMG-TO-IMG>\u0009[OK]"),  
"MiddleFormats":["JPG","JPG"],  
"OutputFileName":"a74a65c5-c012-422f-b8a7-0efd1c2ff237\\testImage.jpg",  
"OutputFullFilePath":"636661428386281277_9\\a74a65c5-c012-422f-b8a7-0efd1c2ff237\\testImage.jpg",  
"ReAttachedToParent":"","  
"Result":2,  
"SourceFileType":"jpg",  
"SourceTrueFileType":"jpeg;jpg",  
"StartJobDateTime":"\\/Date(1530535244000+0300)\\/"}]}
```

Download File

1. General Download

GET v3/download/file/{jobID}

Request Information

URL Example

<https://<Base Address>/Scanner.svc/v3/download/file/05ce1e74-46bd-4f9f-8602-15f9e96090c7>

2. Base64 Download

GET v3/download/file/base64/{jobID}

Request Information

URL Example

<https://<Base Address>/Scanner.svc/v3/download/file/base64/05ce1e74-46bd-4f9f-8602-15f9e96090c7>

URI Parameters

Name	Description	Type	Additional information
JobID	The server Job ID.	String	Required - The Job ID that returned by the engine in the upload phase.

Body Parameters

None.

Request Formats

HTTP Header Parameters

Name	Required\Optional	Description	Type
FileName	Required	Use <code>OutputFileName</code> property from the scan logs.	String
Accept	Required	Value: application/octet-stream	String

Sample:

Sample of header.

```
{
  FileName: 8b98239a-dd07-40a3-822c-bb9d918001ae\\testImage.jpg
  Accept: application/octet-stream
}
```

Response Information

Resource Description

[Stream\Binary](#)

Response Formats

application/json

Sample:

Stream\Binary of file.

Clean Server Data

DELETE v3/delete/{jobID}

Request Information: This action will delete server temporary data if exist, it will not delete the job file or logs in engine.

URL Example

https://<Base Address>/Scanner.svc/v3/delete/05ce1e74-46bd-4f9f-8602-15f9e96090c7

URI Parameters

Name	Description	Type	Additional information
JobID	The Job ID.	String	Required - The Job ID that was given by the engine in the upload phase. Use for deleting data on server.

Body Parameters

None.

Response Information

Resource Description

string

Response Formats

application/json

Sample:

```
"ok"
```

General Resource Description

JobInfo

Name	Description	Type	Additional information
ScansLogArr	Engines scans log.	Collection of ScansLog	The job logs from engine after scan ends.
JobDescription	The Job Description.	String	The Job Description of engine scanning process.
JobID	The server Job ID.	String	The Job ID that was given by the engine at the first upload call. Job ID been given by engine for each job (batch or single) upload.
PlaceInQueue	Place in queue to engine.	Integer	The job place in queue in case multiple jobs are in waiting to be scanned.
ScanProgress	The job scan progress.	Integer	Percentage value for the job while it is running in engine.
ScanStageProgress	Progress of a stage. 50% of stage 2 from 10 stages.	Integer	Not implemented yet.
Scan_Status_c	The Job status.	ScanStatus	Status of job such as InQueue, InScan, Finished, etc'
JobError	Job application error.	String	Application errors with error number.

ScansLog

Name	Description	Type	Additional information
Base64ObjData		String	Optional, depends on engine profile.
Base64ObjName		String	Optional, depends on engine profile.
DestinationFileType	File format after conversion.	String	Optional, depends on engine profile.
DurationSeconds	Conversion time.	String	Optional, depends on engine profile.
EmbeddedLogs	Collection of ScansLog (Only for Embedded files of a converted pdf, docx, etc')	Collection of ScansLog	Optional, depends on engine profile.
EmbeddedParentName	When the file is extracted from an embedded parent file (pdf, doc, etc')	String	Optional, depends on engine profile.
FileHASH	The original file HASH.	String	Optional, depends on engine profile.
FileName	The original filename.	String	N/A.
FileNewHASH	The file HASH after conversion.	String	Optional, depends on engine profile.
FileNewSizeBytes	The file size after conversion.	String	Optional, depends on engine profile.
FileSizeBytes	The original file size.	String	N/A.

Name	Description	Type	Additional information
FullFilePath	The original file path on server.	String	N/A
InnerLogs	Collection of ScanLog (compressed files of a converted zip, rar, archive, etc').	Collection of ScansLog	Optional, depends on engine profile.
IsReAttached	When a file is from an embedded file pdf, docx, etc' was reattached (pdf output only is available).	Boolean	Optional, depends on engine profile.
LogErrorMessages	Collection of errors from engine.	Collection of string	N/A.
LogMessages	Collection of messages from engine.	Collection of string	N/A.
MiddleFormats	Double conversion.	Collection of string	Optional, depends on engine profile.
OutputFileName	The output filename.	String	Needed for Download file request
OutputFullFilePath	The output file path on server.	String	N/A
Result	Enum status representing the engine results: Ok, Reconstructed, Drop etc'.	ScanFileResult	N/A

Name	Description	Type	Additional information
ReAttachedToParent	Parent Name, when a file is from an embedded file pdf, docx etc' was reattached (pdf output only available).	String	Optional, depends on engine profile.
SourceFileType	The file original type.	String	N/A.
SourceTrueFileType	The file true type.	String	N/A.
StartJobDateTime	The scan start timestamp.	Date	N/A.

ScanStatus

Possible enumeration values:

Name	Value	Description
Unknown	0	The server returns unknown status for the job.
InQueue	1	The job is in queue waiting for scan.
InScan	2	The job is now scanning.
Finished	3	The scan ends.
Received	4	The scan ends.

Name	Value	Description
PreScan	5	Returned by the server in case the file is in pre-queue state at engine side.

ScanFileResult

Possible enumeration values:

Name	Value	Description
Ok	0	The scan ended with no findings.
Drop	1	The scan ended with failure. The file was dropped.
Reconstructed	2	The scan ended with findings The file was reconstructed.
Error	3	Error in Application.

Scenarios Samples

Introduction

The Gate Scanner® Rest API in general will include a number of scan scenarios.

- Single non-chunk file process (Upload, Job Status, Download, Delete).
- Single base64 non-chunk file process (Upload, Job Status, Download, Delete).
- Batch upload, scan and download 2 files.
- Batch upload with chunks of big file, scan and download all files.

Bellow a description of the scenarios, include the HTTPS Method (POST\GET), request information and response information.

Single non-chunk file process (Upload, Job Status, Download, Delete)

- **CURL Sample:**

1. **Upload:**

```
curl -i -X POST -H "Content-Type: application/octet-stream" -H "GS-Key: <Issued API Key>" -H "FileName: <Issued Filename>" -H "AppInfo: : <Issued appInfo>" -H "profileID: <Issued ProfileID>" -H "F-Size: <Issued file Size in Bytes>" --data-binary @"file path" https://<Base Address>/Scanner.svc/v3/upload/file
```

2. **Scan Status:**

```
curl -i -X GET https://<Base Address>/Scanner.svc/v3/scan/{attempt}/{jobID}
```

3. **Download File:**

```
curl -i -X GET -H "FileName: <Issued OutputFilename>" -H "Accept: application/octet-stream" https://<Base Address>/Scanner.svc/v3/download/file/{jobID}
```

4. **Delete:**

```
curl -i -X DELETE https://<Base Address>/Scanner.svc/v3/delete/{jobID}
```

- Postman Collection:



REST API Single
non-chunk File.json



sample.pdf

Single base64 non-chunk file process (Upload, Job Status, Download, Delete).

- Postman Collection:



REST API Base64
File.json



sample_pdf.txt

Batch Upload, scan and download 2 files.

Upload file 1: **testDoc.docx**

Request URL

https://<Base Address>/Scanner.svc/v3/upload/file

Request Parameters

None.

Body Parameters

File Stream\Binary.

Request Header

Sample:

Sample of header.

```
{
  FileName: testDoc.docx
  GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92
  AppInfo: AppInfo
  profileID: 4
  F-Size: 10620426
  ScanJob: false
  Accept: application/json
}
```

Server Response

application/json

Sample:

```
{
  "JobDescription":null,
  "JobError":null,
  "JobID":"8b98239a-dd07-40a3-822c-bb9d918001ae",
```

```
"PlaceInQueue":0,"ScanProgress":0,  
"ScanStageProgress":0,"Scan_Status_c":5,  
"ScansLogArr":null  
}
```

Upload file 2: **txtTest.txt**

Request URL

https://<Base Address>/Scanner.svc/v3/upload/file

Request Parameters

None.

Body Parameters

File Stream\Binary.

Request Header

Sample:

Sample of header.

```
{  
  FileName: txtTest.txt  
  GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92  
  AppInfo: AppInfo  
  profileID: 4  
  F-Size: 28  
}
```

```
ScanJob: true
```

```
JobID: 8b98239a-dd07-40a3-822c-bb9d918001ae
```

```
Accept: application/json
```

```
}
```

Server Response

application/json

Sample:

```
{  
  "JobDescription":null,  
  "JobError":null,  
  "JobID":"8b98239a-dd07-40a3-822c-bb9d918001ae",  
  "PlaceInQueue":-99,  
  "ScanProgress":0,  
  "ScanStageProgress":0,  
  "Scan_Status_c":1,  
  "ScansLogArr":null  
}
```

Scan to end, iteration 1

Request URL

https://<Base Address>/Scanner.svc/v3/scan/1/8b98239a-dd07-40a3-822c-bb9d918001ae

Request Parameters

Name	Value	Type	Additional information
attempt	1	String	Required - Increase the value for each call.
JobID	8b98239a-dd07-40a3-822c-bb9d918001ae	String	Required - The Job ID that was given by the engine in the upload phase.

Request Header

None.

Server Response

application/json

Sample:

```
{
  "JobDescription": "DocxsReform 50.00%",
  "JobError": null,
  "JobID": "8b98239a-dd07-40a3-822c-bb9d918001ae",
  "PlaceInQueue": 0,
  "ScanProgress": 45,
  "ScanStageProgress": 0,
  "Scan_Status_c": 2,
  "ScansLogArr": null
}
```

```
}
```

Scan to end, iteration 2

Request URL

https://<Base Address>/Scanner.svc/v3/scan/2/8b98239a-dd07-40a3-822c-bb9d918001ae

Request Parameters

Name	Value	Type	Additional information
attempt	2	String	Required - Increase the value for each call.
jobID	8b98239a-dd07-40a3-822c-bb9d918001ae	String	Required - The Job ID that was given by the engine in the upload phase.

Request Header

None.

Server Response

application/json

Sample:

```
{  
  "JobDescription":null,  
  "JobError":null,  
  "JobID":"8b98239a-dd07-40a3-822c-bb9d918001ae",  
  "PlaceInQueue":0,  
}
```

```
"ScanProgress":0,
"ScanStageProgress":0,
"Scan_Status_c":3,
"ScansLogArr":
[
{
"Base64ObjData":null,
"Base64ObjName":null,
"DestinationFileType":null,
"DurationSeconds":"6.984",
"EmbeddedLogs":null,
"EmbeddedParentName":"","
"FileHASH":"d2f3ea10fb326dac889bce792dc2503c1eab1aa044e9c779f0f2e1cad5dd6a0b",
"FileName":"Doc26.11.docx",
"FileNewHASH":"c338a6e3ffa7355876deea588ea44b4ccb66dd79f5781f63c05b248eeeea44307",
"FileNewSizeBytes":"120741",
"FileSizeBytes":"10620426",
"FullFilePath":"8b98239a-dd07-40a3-822c-bb9d918001ae\\testDoc.docx",
"InnerLogs":null,
"IsReAttached":false,
"LogErrorMessage":null,
"LogMessages":
[
" Pages:2 | Macros:0 | Embedded:0 | Charts:0 | ImagesConverted:5 | Search & Replace Results: | Rule:1 (0) | Rule:2 (0) -> DOCX Reformer Engine - OK","Passed Description : <CONVERTED DOC-TO-MHT, => DOC-DOCX>\u0009[OK]"
],
"MiddleFormats":["MHT","DOC\\DOCX"],
"OutputFileName":"8b98239a-dd07-40a3-822c-bb9d918001ae\\testDoc.docx",
"OutputFullFilePath":"636662267627702644_4\\8b98239a-dd07-40a3-822c-bb9d918001ae\\testDoc.docx",
```

```
"ReAttachedToParent":"","  
"Result":2,  
"SourceFileType":"docx",  
"SourceTrueFileType":"docx;docx",  
"StartJobDateTime":"\\/Date(1530619171000+0300)\\/"  
},  
{  
"Base64ObjData":null,  
"Base64ObjName":null,  
"DestinationFileType":null,  
"DurationSeconds":"0.000000",  
"EmbeddedLogs":null,  
"EmbeddedParentName":"","  
"FileHASH":"461dc44f67b39d564276f448b2018fd309203198ff79f8fdf9e7e7d381095030",  
"FileName":"txtTest.txt",  
"FileNewHASH":"36c2d23557c222c058d649ec747f98ceb975ac37a11a71ee04dbaa75a8c49f38",  
"FileNewSizeBytes":"30",  
"FileSizeBytes":"28",  
"FullFilePath":"8b98239a-dd07-40a3-822c-bb9d918001ae\\txtTest.txt",  
"InnerLogs":null,  
"IsReAttached":false,  
"LogErrorMessages":null,  
"LogMessages":  
[  
"Match 65001 -> TXT Engine - OK","Passed Description : <CONVERTED TXTs-TO-ANY>\\u0009[OK]"  
],  
"MiddleFormats":["TXT"],  
"OutputFileName":"8b98239a-dd07-40a3-822c-bb9d918001ae\\txtTest.txt",  
"OutputFullFilePath":"636662267627702644_4\\8b98239a-dd07-40a3-822c-bb9d918001ae\\txtTest.txt",  
"ReAttachedToParent":"","
```

```
"Result":2,  
"SourceFileType":"txt",  
"SourceTrueFileType":"","  
"StartJobDateTime":"\\/Date(1530619178000+0300)\\/"  
}}}
```

Download file 1: **testDoc.docx**

Request URL

<https://<Base Address>/Scanner.svc/v3/download/file/8b98239a-dd07-40a3-822c-bb9d918001ae>

Request Parameters

Name	Value	Type	Additional information
jobID	8b98239a-dd07-40a3-822c-bb9d918001ae	String	Required - The Job ID that was given by the engine in the upload phase.

Request Header

Sample:

Sample of header.

```
{  
FileName: 8b98239a-dd07-40a3-822c-bb9d918001ae\\testDoc.docx  
Accept: application/octet-stream  
}
```

Server Response

Stream\Binary.

Download file 2: **txtTest.txt**

Request URL

https://<Base Address>/Scanner.svc/v3/download/file/8b98239a-dd07-40a3-822c-bb9d918001ae

Request Parameters

Name	Value	Type	Additional information
jobID	8b98239a-dd07-40a3-822c-bb9d918001ae	string	Required - The Job ID that was given by the engine in the upload phase.

Request Header

Sample:

Sample of header.

```
{  
FileName: 8b98239a-dd07-40a3-822c-bb9d918001ae\\txtTest.txt  
Accept: application/octet-stream  
}
```

Server Response

Stream\Binary.

Batch Upload with chunks of a big file (one out of two files), Scan and Download Scenario.

Upload file 1: **DocMParts.docx**

Part 1

Request URL

<https://<Base Address>/Scanner.svc/v3/upload/file>

Request Parameters

None.

Body Parameters

File Stream (part 1).

Request Header

Sample:

Sample of header.

```
{  
  FileName: DocMParts.docx  
  GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92  
  AppInfo: AppInfo  
  profileID: 6  
  F-Size: 10620426
```

```
ScanJob: false
```

```
Accept: application/json
```

```
}
```

Server Response

application/json

Sample:

```
{  
  "JobDescription":null,  
  "JobError":null,  
  "JobID":"a8daa307-b95a-4274-834c-a69a261bed6d",  
  "PlaceInQueue":0,  
  "ScanProgress":0,  
  "ScanStageProgress":0,  
  "Scan_Status_c":5,  
  "ScansLogArr":null  
}
```

Part 2

Request URL

https://<Base Address>/Scanner.svc/v3/upload/file

Request Parameters

None.

Body Parameters

File Stream\Binary (part 2).

Request Header

Sample:

Sample of header.

```
{
  FileName: DocMParts.docx
  GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92
  AppInfo: AppInfo
  profileID: 6
  F-Size: 10620426
  JobID: a8daa307-b95a-4274-834c-a69a261bed6d
  ScanJob: false
  Accept: application/json
}
```

Server Response

application/json

Sample:

```
{
  "JobDescription":null,
  "JobError":null,
  "JobID":"a8daa307-b95a-4274-834c-a69a261bed6d",
  "PlaceInQueue":0,
  "ScanProgress":0,
  "ScanStageProgress":0,
  "Scan_Status_c":5,
  "ScansLogArr":null
}
```

Part 3

Request URL

https://<Base Address>/Scanner.svc/v3/upload/file

Request Parameters

None.

Body Parameters

File Stream\Binary (part 3, last part).

Request Header

Sample:

Sample of header.

```
{
  FileName: DocMParts.docx
  GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92
  AppInfo: AppInfo
  profileID: 6
  F-Size: 10620426
  JobID: a8daa307-b95a-4274-834c-a69a261bed6d
  ScanJob: false
  Accept: application/json
}
```

Server Response

application/json

Sample:

```
{
```

```
"JobDescription":null,  
"JobError":null,  
"JobID":"a8daa307-b95a-4274-834c-a69a261bed6d",  
"PlaceInQueue":0,  
"ScanProgress":0,  
"ScanStageProgress":0,  
"Scan_Status_c":5,  
"ScansLogArr":null  
}
```

Upload file 2: **ScanTest.txt**

Request URL

https://<Base Address>/Scanner.svc/v3/upload/file

Request Parameters

None.

Body Parameters

File Stream\Binary.

Request Header

Sample:

Sample of header.

```
{  
FileName: ScanTest.txt  
GS-Key: 240ae63c-3259-4438-b60b-7a8a54909b92  
AppInfo: AppInfo  
profileID: 6
```

F-Size: 28

ScanJob: true

JobID: a8daa307-b95a-4274-834c-a69a261bed6d

Accept: application/json

}

Server Response

application/json

Sample:

```
{
  "JobDescription":null,
  "JobError":null,
  "JobID": " a8daa307-b95a-4274-834c-a69a261bed6d",
  "PlaceInQueue": -99,
  "ScanProgress":0,
  "ScanStageProgress":0,
  "Scan_Status_c":1,
  "ScansLogArr":null
}
```

Scan to end, iteration 1

Request URL

https://<Base Address>/Scanner.svc/v3/scan/1/a8daa307-b95a-4274-834c-a69a261bed6d

Request Parameters

Name	Value	Type	Additional information
attempt	1	String	Required - Increase the value for each call.
JobID	a8daa307-b95a-4274-834c-a69a261bed6d	String	Required - The Job ID that was given by the engine in the upload phase.

Request Header

None.

Server Response

application/json

Sample:

```
{
  "JobDescription": "Extract Documents Embeddeds ",
  "JobError": null,
  "JobID": "a8daa307-b95a-4274-834c-a69a261bed6d",
  "PlaceInQueue": 0,
  "ScanProgress": 17,
  "ScanStageProgress": 0,
  "Scan_Status_c": 2,
  "ScansLogArr": null
}
```

Scan to end, iteration 2

Request URL

https://<Base Address>/Scanner.svc/v3/scan/2/a8daa307-b95a-4274-834c-a69a261bed6d

Request Parameters

Name	Value	Type	Additional information
attempt	2	String	Required - Increase the value for each call.
jobID	a8daa307-b95a-4274-834c-a69a261bed6d	String	Required - The Job ID that was given by the engine in the upload phase.

Request Header

None.

Server Response

application/json

Sample:

```
{  
  "JobDescription":null,  
  "JobError":null,  
  "JobID":"a8daa307-b95a-4274-834c-a69a261bed6d",  
  "PlaceInQueue":0,  
  "ScanProgress":0,  
  "ScanStageProgress":0,  
  "Scan_Status_c":3,  
  "ScansLogArr":
```

```
[
{
"Base64ObjData":null,
"Base64ObjName":null,
"DestinationFileType":null,
"DurationSeconds":"6.656",
"EmbeddedLogs":null,
"EmbeddedParentName":"","
"FileHASH":"d2f3ea10fb326dac889bce792dc2503c1eab1aa044e9c779f0f2e1cad5dd6a0b",
"FileName":"DocMParts.docx",
"FileNewHASH":"4ea8869d760a9c729c4404aa8cc76bb624c46ada002a398a2cefc551b52c7424",
"FileNewSizeBytes":"120740",
"FileSizeBytes":"10620426",
"FullFilePath":"a8daa307-b95a-4274-834c-a69a261bed6d\\DocMParts.docx",
"InnerLogs":null,
"IsReAttached":false,
"LogErrorMessages":null,
"LogMessages":
[
" Pages:2 | Macros:0 | Embedded:0 | Charts:0 | Hyperlinks:0 | ImagesConverted:5 -> DOCX Reformer Engine - OK","Passed Description : <CONVERTED DOC-TO-MHT, =>DOCX>\u0009[OK]"
],
"MiddleFormats":["MHT","DOCX"],
"OutputFileName":"a8daa307-b95a-4274-834c-a69a261bed6d\\DocMParts.docx",
"OutputFullFilePath":"636662950623407636_6\\a8daa307-b95a-4274-834c-a69a261bed6d\\DocMParts.docx",
"ReAttachedToParent":"","
"Result":2,
"SourceFileType":"docx",
"SourceTrueFileType":"docx",
"StartJobDateTime":"\\/Date(1530687468000+0300)\\/"
```

```
},  
{  
  "Base64ObjData":null,  
  "Base64ObjName":null,  
  "DestinationFileType":null,  
  "DurationSeconds":"0.000000",  
  "EmbeddedLogs":null,  
  "EmbeddedParentName": "",  
  "FileHASH":"461dc44f67b39d564276f448b2018fd309203198ff79f8fdf9e7e7d381095030",  
  "FileName":"ScanTest.txt",  
  "FileNewHASH":"36c2d23557c222c058d649ec747f98ceb975ac37a11a71ee04dbaa75a8c49f38",  
  "FileNewSizeBytes":"30",  
  "FileSizeBytes":"28",  
  "FullFilePath":"a8daa307-b95a-4274-834c-a69a261bed6d\\ScanTest.txt",  
  "InnerLogs":null,  
  "IsReAttached":false,  
  "LogErrorMessage":null,  
  "LogMessages":  
  [  
    "Match 65001 -> TXT Engine - OK","Passed Description : <CONVERTED TXTs-TO-ANY>\u0009[OK]"  
  ],  
  "MiddleFormats":["TXT"],  
  "OutputFileName":"a8daa307-b95a-4274-834c-a69a261bed6d\\ScanTest.txt",  
  "OutputFullFilePath":"636662950623407636_6\\a8daa307-b95a-4274-834c-a69a261bed6d\\ScanTest.txt",  
  "ReAttachedToParent": "",  
  "Result":2,  
  "SourceFileType":"txt",  
  "SourceTrueFileType": "",  
  "StartJobDateTime":"\\/Date(1530687475000+0300)\\/"  
}]}
```

Download file 1: **DocMParts.docx**

Request URL

https://<Base Address>/Scanner.svc/v3/download/file/a8daa307-b95a-4274-834c-a69a261bed6d

Request Parameters

Name	Value	Type	Additional information
jobID	a8daa307-b95a-4274-834c-a69a261bed6d	string	Required - The Job ID that was given by the engine in the upload phase.

Request Header

Sample:

Sample of header.

```
{  
FileName: a8daa307-b95a-4274-834c-a69a261bed6d\DocMParts.docx  
Accept: application/octet-stream  
}
```

Server Response

Stream\Binary.

Download file 2: **ScanTest.txt**

Request URL

https://<Base Address>/Scanner.svc/v3/download/file/a8daa307-b95a-4274-834c-a69a261bed6d

Request Parameters

Name	Value	Type	Additional information
JobID	a8daa307-b95a-4274-834c-a69a261bed6d	String	Required - The Job ID that was given by the engine in the upload phase.

Request Header

Sample:

Sample of header.

```
{  
FileName: a8daa307-b95a-4274-834c-a69a261bed6d\\ScanTest.txt  
Accept: application/octet-stream  
}
```

Server Response

Stream\Binary.