Sasa Software
Smart Made Simple.

GATESCANNER INTEGRATION SERVER

API Document

Your gate to a safe system

Version 2.0.1.1

Document History

Version Number	Version Date	Author	Product Version
1	November 2023	Dotan Amor	2.0.1.1

Table of Contents

1. INTRODUCTION 4
2. HEALTH CHECK 5
2.1. Request 5
2.2. Response 5
3. OPEN NEW JOB 6
3.1. Request 6
3.2. Response 6
.4 UPLOAD FILE (CIRCULAR REQUEST) 7
4.1. Binary 7
4.1.1. Request 7
4.1.2. Response 7
4.2. Base64 7
4.2.1. Request 7
4.2.2. Response 7
4.3. Code Sample C# 8
5. ACTIVATE JOB 10
5.1. Request 10
5.2. Response 10
6. SCAN STATUS (JOB PROGRESS) 11
6.1. Request 11
6.2. Response 11
7. DOWNLOAD FILES 12
7.1. Binary 12
7.1.1. Request 12
7.1.2. Response 12
7.1.3. Developer Notes 12
7.1.4. Reconstructed Note 12
7.1.5. Dummy Files Note 12
7.2. Base64 13
7.2.1. Request 13
7.2.2. Response 13
7.3. Sample Code C# 13
8. DELETE 16
8.1. Request 16
8.2. Response 16
9. PASSWORD PROTECTED FILES 17
10. DUMMY FILES 18
10.1. Admin site basic Configuration 18
11. ERROR MESSAGES 19
12. CONTACT INFORMATION 22

1. Introduction

This document is intended for use by programmers wishing to connect to the GateScanner Integration Server. It outlines expected requests and responses when using the different abilities of the Integration Server, using RESTful API.

2. Health Check

2.1. Request

URL: https://<server IP>:<port>/v4/hello
Method: GET
Headers: none.
Body: none.

2.2. Response

Status Code: 200
Content: "i am alive".

Any other response is an error.

3. Open New Job

3.1. Request

URL: https://<server IP>:<port>/v4/OpenJob
Method: POST

Headers:

- GS-Key: a unique identifier (GUID from AdminRest).
- BatchScan_DeclaredTotalJobsSizeBytes: total size in bytes.
- BatchScan_DeclareFileCount: number of files in batch.

Body: none.

3.2. Response

- Object type 'JobInfo', containing the **JobID** for future use (save it).
- If an error occurs, '**JobError'** will contain the error code and '**JobDescription'** will contain a short description of the error.

When done, continue to Upload File.

4. Upload File (circular request)

4.1. Binary

4.1.1. Request

URL: https://<server IP>:<port>/v4/ upload/file

Method: POST

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).
- FileName: the file name to upload.
- F-Size: file size in bytes (total size).
- IsLastFilePart: indicate the last packet of the file.
- AppInfo: external client identifier.

Body: file content as binary.

4.1.2. Response

If an error occurs, '**JobError'** will contain the error code and '**JobDescription'** will contain a short description of the error.

When done uploading, continue to Activate the Job.

4.2. Base64

4.2.1. Request

URL: https://<server IP>:<port>/v4/ uploadBase64/file

Method: POST

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).
- FileName: the file name to upload.
- F-Size: file size in bytes (total size).
- IsLastFilePart: indicate the last packet of the file.
- AppInfo: external client identifier.

Body: file content base64 as binary.

File chunk/part: In case of dividing the payload across multiple requests, send the payload evenly divided to 8 or 6 parts.

4.2.2. Response

If an error occurs, '**JobError'** will contain the error code and '**JobDescription'** will contain a short description of the error.

4.3. Code Sample C#

```
try
{
bool isJobIdMsgSubmit = false;
JobInfo jobInfo = null;
string fileName = Path.GetFileName(fullPath);
_logger.LogWriter(AppLogger.Severity.Info, this.GetType().FullName, "UploadInPartsV2()", " ", "Data: ", "File Name for Scan: " + fileName);
Enqueue(_appMessages, "File Name for Scan: " + fileName);
Enqueue(_submittedFile, "Start Process For File: " + fileName);
int lengthBuffer = (this.ChunkSize);
string data = string.Empty;
using (BinaryReader reader = new BinaryReader(File.Open(fullPath, FileMode.Open, FileAccess.Read, FileShare.Read)))
{
// create a buffer to hold the bytes
byte[] buffer = new Byte[lengthBuffer];
int bytesRead;
int CurrentFilePart = 0;
while ((bytesRead = reader.Read(buffer, 0, lengthBuffer)) > 0)
{
CurrentFilePart++;
bool IsLastFilePart = reader.BaseStream.Position >= reader.BaseStream.Length;
if (IsLastFilePart)
{
Array.Resize<byte>(ref buffer, bytesRead);
}
jobInfo = UploadPartV4(fullPath, buffer, gsKey, fileSize, jobID, IsLastFilePart);
if (IsLastFilePart)
{
try
{
Enqueue(_submittedFile, "JobID: " + jobID + ", Last Part Part: " + CurrentFilePart.ToString());
}
catch (Exception)
{
}
```

```
            break;
          }
          jobID = jobInfo.JobID;
          if (!isJobIdMsgSubmit)
          {
            Enqueue(_submittedFile, "JobID Connection: " + jobInfo.JobID);
            isJobIdMsgSubmit = true;
          }
          try
          {
            Enqueue(_submittedFile, "JobID: " + jobID + ", Part: " + CurrentFilePart.ToString());
          }
          catch (Exception)
          {
          }
          if (!string.IsNullOrEmpty(jobInfo.JobError))
          {
            Enqueue(_submittedFile, "JobID: " + jobID + ", Part: " + CurrentFilePart.ToString() + " operation stoped due to
error in engine response.");
            break;
          }
        }
      }
      return jobInfo;
    }
    catch (Exception ex)
    {
      throw ex;

    }
```

When done uploading, continue to Activate the Job.

5. Activate Job

5.1. Request

URL: https://<server IP>:<port>/v4/ ActivateJob/{JobID}
Method: POST

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).
- profileID: GateScanner profileID (from MAM).
- AppInfo: user data (can be passed as a Hardcoded string from the ICAP Server).
- F-Password: serialize array of strings e.g., "F-Password":"[\"123456\"]".
  See more about password-protected files in the dedicated section below.

Body: file content as binary.

5.2. Response

If an error occurs, '**JobError'** will contain the error code and '**JobDescription'** will contain a short description of the error.

When Activate Job is complete, continue to the next section.

6. Scan Status (Job Progress)

6.1. Request

URL: https://<server IP>:<port>/v4/ scan/{iterator}/{JobID}
Method: GET

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).

Body: none.

6.2. Response

- If an error occurs, '**JobError'** will contain the error code and '**JobDescription'** will contain a short description of the error.
- Object of type 'JobInfo', containing the **Scan_Status_c**.
- **Scan_Status_c** is part of an Enum structture called **ScanStatus** –
    - Unknown = 0
    - InQueue = 1
    - InScan = 2
    - Finished = 3
    - Received = 4
    - PreScan = 5
    - InitialValue = 6
    - UploadToEngine = 7
    - DownloadFromEngine = 8
    - InConstruction = 9
- Use a switch statement to detect the API response with the engine scan progress and result:
    - **0** - when an error occurs more than once, return error.
    - **1,2,5,7,9** - wait 3 seconds and check again (preferably configure a parameter to control the interval).
    - **3 and 4** - scan complete, inspect the **ScansLogArr** property –
        - If the array contains objects, each object is a file.
        - If the file contains attachments or embedded files, the details can be found in **InnerLogs** OR **EmbeddedLogs**.
- If the first element in the array contains a property called **Result** - Result is also an Enum structure called **ScanFileResult**:
    - Ok = 0
    - Drop = 1
    - Reconstructed = 2
    - Error = 3

When the scan is complete, move to the next section.

7. Download Files

7.1. Binary

7.1.1. Request

URL: https://<server IP>:<port>/v4/ download/file/{JobID}
Method: GET.

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).
- FileName: from ScansLogArr - OutputFileName.
- MaxReadBytes: the default is 256*1024 bytes (2MB), if the file is bigger multiple requests are required until the file is fully downloaded (property IsEOF == true).
- Position (optional): resume download, default value is ZERO.

7.1.2. Response

Body: file content as binary.

- Object of type 'FilePartV4' containing the payload property.
  - payload = raw file data as Byte array.
- IsEOF = indicate if file End (stop requesting content from API).
- startPosition = contains the start position of the current part.
- exception = .Net exception object if any exception occurs. If an error occurs, the **exception property** will contain the error details.

7.1.3. Developer Notes

The Download method will start from position 0 and will receive in response the updated value in the property **startPosition**.

7.1.4. Reconstructed Note

when **ScanFileResult** is returned with value 2, check the following:

1. the new file size - to be saved on disk.
2. the new file extension - from **ScansLogArr** OutputFileName (may contain multiple extensions).
3. FileNewHash - if activated in the GateScanner engine, this value will be different than FileHash (original) value.

7.1.5. Dummy Files Note

When ScanFileResult is returned with value 1 and Dummy Files Feature (in the admin site) is enabled, a download request will return the dummy file: <name of file>.err.txt.
The reason for the file removal is configured in the admin site to be saved in the file content.

When download is complete, continue to the next section.

7.2. Base64

7.2.1. Request

URL: https://<server IP>:<port>/v4/ downloadBase64/file/{JobID}

Method: GET.

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).
- FileName: from ScansLogArr OutputFileName.
- MaxReadBytes: the default is 256*1024 bytes (2MB), if the file is bigger multiple requests are required until the file is fully downloaded (property IsEOF == true).
- Position (optional): resume download, default value is ZERO.

7.2.2. Response

Body: file content as binary.

- Object of type 'FilePartV4' containing the payload property.
  - payload = raw file data as Byte array.
- IsEOF = indicate if file End (stop requesting content from API).
- startPosition = contains the start position of the current part.
- exception = .Net exception object if any exception occurs. If an error occurs, the **exception property** will contain the error details.

7.3. Sample Code C#

```
try
    {
     _logger.LogWriter(AppLogger.Severity.Info, this.GetType().FullName, stackFrame.GetMethod().Name, ", url: " + url.ToString(), " ;method:" + method.ToString(), ", url: " + url.ToString() + " ;method: " + method.ToString() + ", localPath: " + localPath.ToString());
     // Set a default policy level for the "http:" and "https" schemes.
     HttpRequestCachePolicy policy = new HttpRequestCachePolicy(HttpRequestCacheLevel.NoCacheNoStore);
     HttpWebRequest.DefaultCachePolicy = policy;
     HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
     request.Method = method;
     if (headers != null)
     {
      foreach (KeyValuePair<string, string> pair in headers)
      {
       _logger.LogWriter(AppLogger.Severity.Info, this.GetType().FullName, stackFrame.GetMethod().Name, ", Header: ", " ;pair.Key:" + pair.Key.ToString(), " Header: ;pair.Key:" + pair.Key.ToString() + ", pair.Value: " + pair.Value.ToString());
       request.Headers.Add(pair.Key, pair.Value);
```

```
        }
      }
      request.Accept = "application/octet-stream";
      request.Timeout = System.Int32.MaxValue;

      HttpWebResponse myHttpWebResponse = (HttpWebResponse)request.GetResponse();

      _logger.LogWriter(AppLogger.Severity.Info, this.GetType().FullName, stackFrame.GetMethod().Name, '', url: '' +
url.ToString(), '' ;method:'' + method.ToString(), '', url: '' + url.ToString() + '' ;method: '' + method.ToString() + '', localPath: '' +
localPath.ToString() + '', IsFromCache? '' + myHttpWebResponse.IsFromCache.ToString());
      string results = string.Empty;
      // Gets the stream associated with the response.
      using (StreamReader streamReader = new StreamReader(myHttpWebResponse.GetResponseStream()))
      {
        results = streamReader.ReadToEnd();
      }
      string res = results.Replace("\"", " ").Replace(@"\/", "/").Trim();
      File.WriteAllBytes(localPath, Convert.FromBase64String(res));
      myHttpWebResponse.Close();
      // Releases the resources of the Stream.
      request = null;
      myHttpWebResponse = null;
      return true;
    }
    catch (WebException ex)
    {
      if (ex.Status == WebExceptionStatus.ProtocolError && ex.Response != null)
      {
        _logger.LogWriter(AppLogger.Severity.Error, this.GetType().FullName, stackFrame.GetMethod().Name, '', Source: '' +
ex.Source, '' ;Stack Trace:'' + ex.StackTrace, '', Message: '' + ex.Message);
        HttpWebResponse response = ex.Response as HttpWebResponse;
        StreamReader streamReader = new StreamReader(ex.Response.GetResponseStream());
        int responseReaderCode = (int)response.StatusCode;
        string responseReader = streamReader.ReadToEnd();
        streamReader.Close();
        streamReader.Dispose();
        response = null;
```

```
    return false;
  }
  else
  {
    throw new Exception("Cannot connect to: " + url + " " + ex.Message);
  }
}
catch (UriFormatException ex)
{
  throw new Exception("Cannot connect to: " + url + " " + ex.Message);
}
```

8. Delete

8.1. Request

URL: https://<server IP>:<port>/v4/ delete /{JobID}
Method: DELETE.

Headers:

- GS-Key: unique identifier (GUID from AdminRest).
- JobID: from the Response in 'Open New Job' (see above).

Body: none.

8.2. Response

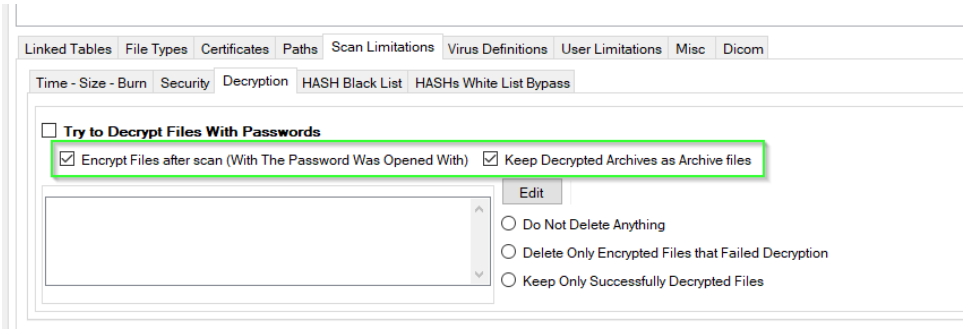
If an error occurs, '**JobError'** will contain the error code and '**JobDescription'** will contain a short description of the error.

9. Password Protected Files

The API allows you to send password protected files to the Integration Server.

Best Practice:

- We recommend clients to send only **One** File with a password for each job.
- GateScanner Engine configuration is required for password protected files.



- The password will be sent in the header parameter in the Activate Job.

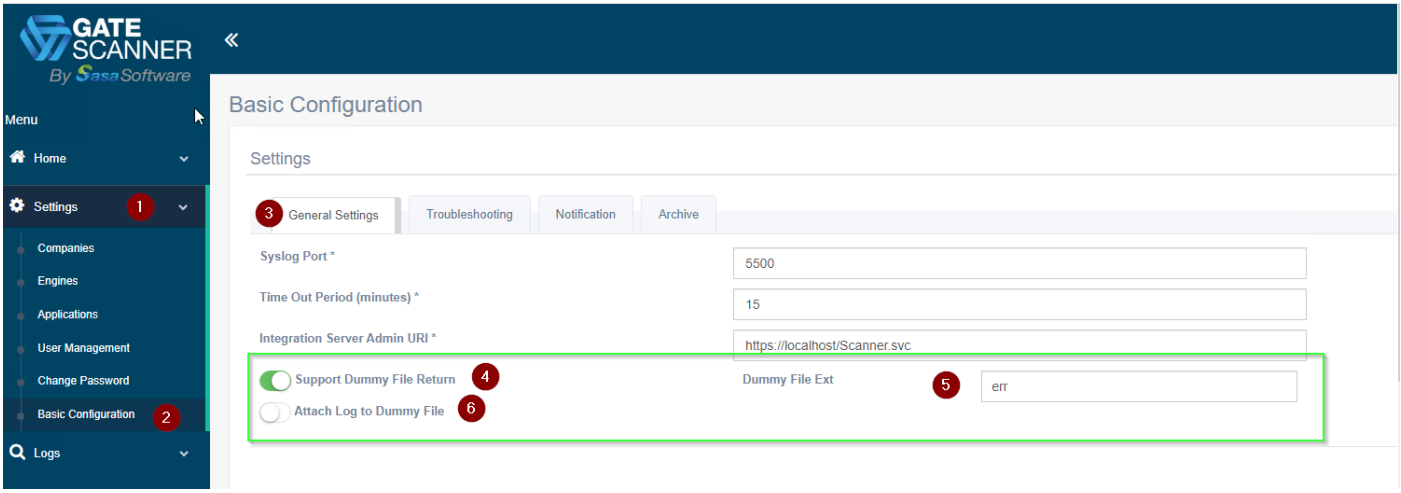
10. Dummy Files

This feature allows the client to receive a file from a Download Request even if the file was removed by the GateScanner Engine.

The return file can be customized to your own File Extension Error policy and can include the reason of removal as file content.

10.1. Admin site basic Configuration

For dummy file download, use the following configuration in the admin site:



11. Error Messages

The table below describes the Integration Server's Error Messages.

Error	Description
565,566	The Header **jobID** is not valid.
567	Header with password, the password is invalid.
568	Header **JobID** and URL **JobID** aren't matched.
601	IO exception occurred, please see log for more details.
1000	The Incoming Web Request Context is null.
1001, 1002	The Web Header Collection is null. The Web Header Collection is empty.
1003	Call with empty GS Key in header is not allowed.
1005	The operation failed.
1007	Upload operation failed, please check the connectivity with the system engines.
1009,1010,1013	The GS Key Failed to load data.
1015	This call expects true or false in **isScan** header, the call is not valid.
1016, 1017,5132	Failed to load **jobInfo**, server internal error.
1020	Upload file action: Incoming part (bytes) too big, the default is **1073741824(Bytes)** in config.
1021, 1022	No active engines found for application.
1023	files are too big - Increase engine 'max single file size' or increase engine HDD capacity.

1026,1027	This call expects to return with scan Info records, the scan Info is null. This call expects to return with scan Info records, the scan Info is empty.
1113	Upload file action, Header with **FileName** length > **4000** is not allowed.
3021	Upload file action: The **multipart file** is larger than declared.
3022	The **file** is larger than declared.
3023	`**IsLastPart**` Header Param must be declared to true when the last packet of a file is sent.
3024	`**IsLastPart**` Header Param is true but the file size is smaller than declared.
5011	Upload file action: internal server error.
5112	The data directory at server does not exist.
5114	Upload file action: internal server error with settings. Might be caused by a lack of disk space, a missing local directory or a write permission problem.
5135	Scan status action: cannot find a company related to the application, internal server error.
5136, 5137	Upload file action: Failed to create connection with engine.
5138	Upload file action: failed to create connection with engine, queue too long.
5139	Upload file action: failed to create connection with engine, general error.
5140, 5141, 5142	Upload file action: No engines found related to the application and GS-Key (API Key).

5143	Cannot find file Path for **JobID** and **FileName**, The Server Didn't Get a Call with Valid Information.
5145	Action Scan Job return with false value.
6000	Activation fails due to incorrect files count. The declared @declareFileCount doesn’t match **FilesOnDisk**: @FilesOnDisk.
6001	Activation fails due to incorrect files size. The declared @declareFileSize doesn’t match **FilesOnDisk**: @FilesSizeOnDisk.
6002	Download file with `Dropped` status. File Name: @FileName CDR Results: @cdrResults.

12. Contact Information

Sasa Software Ltd.

Kibbutz Sasa

M.P. Merom Hagalil 1387000

Israel

Telephone: +972 4 8679959

Fax: +972 4 6918876

Email: info@sasa-software.com

Web: www.sasa-software.com