Sasa Software
Smart Made Simple.

GATE SCANNER
Security Dome
Development API Guide

Your gate to a safe system

Revision 1.0.22

July 2021

Sasa Software (C.A.S) 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

Introduction

Thank you for choosing Gate Scanner® Security Dome (GSSD) 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.

This document describes the GSSD Development API. The Development API enables an integration with the GSSD and use the GSSD functionality using external applciations.

This document is intended for software developers who will be using the GSSD development API. This document provides the public interfaces necessary to use the features provided by the GSSD. A functional overview and information on leveraging the interface functionality are also provided. This document assumes that the user is familiar with web programming and specialy with JS, AJAX and HTTP requests.

Document History

Version	Version Date	Author	Revision & Related Version
1.0.22	July 2021	Kobi Danone	v22

Table of Contents

Development APIs

GetHello

Description

Receives the simplest acknowledge from the server. The agent sends a simple request for Hello and the server responding hello.

URL Format

https://<Base Address>/api/GSPortal/GetHello

Response

Data Type	Value
String	"HELLO FROM API SERVER"

JS Code Example

```
1.    var settings = {
2.      "url": "https://<Base Address>/api/GSPortal/GetHello",
3.      "method": "GET"
      };
4.
5.    $.ajax(settings).done(function (response) {
6.      console.log(response);
7.    });
8.
9.    //Response: HELLO FROM API SERVER
```

GetGlobalValues

Description

Returns a variety of general definitions.

URL Format

https://<Base Address>/api/GSPortal/GetGlobalValues

Response

Data Type	Value
Json/Dictonary	[DefaultSafeReplyCapacityMB, MaxSendSizeMB, DefaultSafeReplyMaxFiles, DefaultSafeReplyMaxReplies, SMSAvilable, EmailAvilable, ForceLogout, SecureFileResponse, TimeoutQueue, DefaultShareDaysLimit, DefaultShareDownloadLimit, MaxShareDaysLimit, MaxShareDownloadLimit, CompactUploadProfile, CompactUploadDestination, Proxy, ProxyAddress]

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/GSPortal/GetGlobalValues",
  "method": "GET"
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response:
/*CompactUploadDestination: "Vault"
CompactUploadProfile: "Default_Alias"
DefaultSafeReplyCapacityMB: "50"
DefaultSafeReplyMaxFiles: "100"
DefaultSafeReplyMaxReplies: "10"
DefaultShareDaysLimit: "4"
DefaultShareDownloadLimit: "3"
EmailAvilable: "True"
ForceLogout: "False"
MaxSendSizeMB: "50"
MaxShareDaysLimit: "7"
MaxShareDownloadLimit: "30"
Proxy: "False"
ProxyAddress: "https://<Proxy Base Address>/"
SMSAvilable: "True"
SecureFileResponse: "True"
TimeoutQueue: "3" */
```

PostReceiveExtAPIToken

Description

Receive Development API key using user and password.
The API key being used to login the GSSD.
The API key remains constant till erased or manually refreshed by the GSSD administrator.

URL Format

https://<Base Address>/api/GSPortal/PostReceiveExtAPIToken

Body Parameters

Data Type	Name	Example
String	Password	“my_password”

Header Parameters

Data Type	Name	Example
String	domain	“local”
String	user	“user1”
GUID	captchaID	“123e4567-e89b-12d3-a456-426614174000” or empty if no need
String	captchaCode	“123aB” or empty if no need
String	factorCode	“123456” or 0 if no need
String	ExtApp	“DefaultDomeApplication” or your External Application Name

Response

Data Type	Value
String	“123e4567-e89b-12d3-a456-426614174123” API key

More Response Options

Type	Value	Description
Captcha	"ERROR:Wrong CaptchaGSSDKEYGSSD" + captchaAnswerBit + "GSSDKEYGSSD" + captchaAnswerID	After 3 failed tries, return base64 captcha image + GUID id.
2FA	1. TwoFactorSMS: XXXX1234 2. TwoFactorEmail:*******@*****.***	Two Factor autentcation via SMS or Email, re-send the login with the 2FA code.

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/GSPortal/PostReceiveExtAPIToken",
  "method": "POST",
  "headers": {
    "user": "user1",
    "domain": "local",
    "captchaID": "",
    "captchaCode": "",
    "factorCode": "0",
    "ExtApp": "DefaultDomeApplication",
    "Content-Type": "application/json"
  },
  "data": JSON.stringify("my_password"),
};

$.ajax(settings).done(function (response) {
  console.log(response);
});

//Response:
/*123e4567-e89b-12d3-a456-426614174123*/
```

PostExtLogin

Description

Login the GSSD using the Development API key and receive a session token to use for most requests.

A session token is valid from the last time the user used the site untill the site Time Out configuration.

Note: The same session token can be used for multiple requests, no need to refresh it in every request.

URL Format

https://<Base Address>/api/GSPortal/PostExtLogin

Body Parameters

Data Type	Name	Example
GUID	Development API Key	“123e4567-e89b-12d3-a456-426614174000”

Header Parameters

Data Type	Name	Example
GUID	captchaID	“123e4567-e89b-12d3-a456-426614174000” or empty if no need
String	captchaCode	“123aB” or empty if no need
String	factorCode	“123456” or 0 if no need
String	ExtApp	“DefaultDomeApplication” or your External Application Name

Response

Data Type	Value
String	“session token, profilePicPath, privilege (1 admin, 2 read admin, 3 domain admin, 4 user), timeOutPeriodMS, EmailServerAvailable, userEmail, isQuotaMode, domainIsAD, allowSecureEmail, allowSecureEmailDesign, SecureEmailCharLimit, allowShareNote, allowShareNoteDesign, ShareNoteCharLimit, DefaultShareDaysLimit, DefaultShareDownloadLimit, MaxShareDaysLimit, MaxShareDownloadLimit, isCompactUploadMode”

More Response Options

Type	Value	Description
Captcha	"ERROR:Wrong CaptchaGSSDKEYGSSD" + captchaAnswerBit + "GSSDKEYGSSD" + captchaAnswerID	After 3 failed tries, return base64 captcha image + GUID id.
2FA	3. TwoFactorSMS: XXXX1234 4. TwoFactorEmail:*******@*****.***	Two Factor autentcation via SMS or Email, re-send the login with the 2FA code.

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/GSPortal/PostExtLogin",
  "method": "POST",
  "headers": {
    "captchaID": "",
    "captchaCode": "",
    "factorCode": "0",
    "ExtApp": "DefaultDomeApplication",
    "Content-Type": "application/json"
  },
  "data": JSON.stringify("123e4567-e89b-12d3-a456-426614174000"),
};

$.ajax(settings).done(function (response) {
  console.log(response);
});

//Response:
/*123e4567-e89b-12d3-a456-426614174123, images\\users\\user.jpg, 4, 1800000, True,
<User Email>, True, False, True, True, 10000, True, False, 300, 4, 3, 7, 30, False*/
```

GetTokenActive

Description

Checks if a session token is still active.

URL Format

https://<Base Address>/api/GSPortal/GetTokenActive

Header Parameters

Data Type	Name	Example
GUID	token	“123e4567-e89b-12d3-a456-426614174000”

Response

Data Type	Value
String	“OK”

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/GSPortal/GetTokenActive",
  "method": "GET",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123"
  },
};

$.ajax(settings).done(function (response) {
  console.log(response);
});

//Response: OK
```

PostLogOut

Description

Logout and close the session token .

URL Format

https://<Base Address>/api/GSPortal/PostLogOut/{onlySetAbility}

URI Parameters

Data Type	Name	Description
Boolean	OnlySetAbility	False – Close the session. True – (Admins only) release the write permission for this session.

Header Parameters

Data Type	Name	Example
GUID	token	“123e4567-e89b-12d3-a456-426614174000”

Response

Data Type	Value
String	“true”

JS Code Example

```
1.    var settings = {
2.      "url": "https://<Base Address>/api/GSPortal/PostLogOut/false",
3.      "method": "POST",
4.      "headers": {
5.        "token": "123e4567-e89b-12d3-a456-426614174123"
6.      },
7.    };
8.
9.    $.ajax(settings).done(function (response) {
10.     console.log(JSON.parse(response));
11.   });
12.
13.   //Response: true
```

GetUserRules

Description

Checks if a session token is still active.

URL Format

https://<Base Address>/api/ApiRule/GetUserRules

Header Parameters

Data Type	Name	Example
GUID	token	“123e4567-e89b-12d3-a456-426614174000”

Response

Data Type	Value
JSON/Struct	{List<string> profileDescription; List<string> profileAlias; String profile; String capacity; String capacityUsed; String scanCapacity; String scanCapacityUsed; String avialble; String MaxDaysInStorage; String MaxFilesInJob; String MaxSingleFileSize; String MaxTotalFilesSizeInJob; String FileTypeBlackList; String FileTypeWhiteList}

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/ApiRule/GetUserRules",
  "method": "GET",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123"
  },
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response:
/*FileTypeBlackList: "Unlimited" FileTypeWhiteList: "Unlimited"
MaxDaysInStorage: "Unlimited" MaxFilesInJob: "Unlimited"
MaxSingleFileSize: "Unlimited" MaxTotalFilesSizeInJob: "Unlimited"
avialble: "Unlimited" capacity: "Unlimited" capacityUsed: "22.3396415710449"
profile: "1,22,3,-1"
profileAlias: (4) ["Default_Alias", "reconstructions", "fastprofile", "Engine Bypass"]
profileDescription: (4) ["Default Engine Profile", "", "", "Engine Bypass"]
scanCapacity: "Unlimited" scanCapacityUsed: "2894.74430274963"*/
```

GetDestinations

Description

Get a list of the destinations (beside the default vault).

URL Format

https://<Base Address>/api/Destinations/GetDestinations/{all}

URI Parameters

Data Type	Name	Description
Boolean	all	False – Get the destination available to the user. True – (Admins only) Get all the destinations.

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”

Response

Data Type	Value
JSON/DataTable	[Alias, AllowDelete, AllowShare, AllowUpload, DestinationType, DestinationUser, IsShared, IsUNCVault]

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/Destinations/GetDestinations/false",
  "method": "GET",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123"
  },
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response:
/*[{Alias: "UNCDest" AllowDelete: true AllowShare: true AllowUpload: true
DestinationType: "unc" DestinationUser: "Scanner" IsShared: false IsUNCVault: false},
{Alias: "myFTPS" AllowDelete: true AllowShare: true AllowUpload: true
DestinationType: "ftps" DestinationUser: "FTPUser" IsShared: false IsUNCVault: false}]*/
```

PostUploadFiles

Description

Upload files to the portal.

URL Format

https://<Base Address>/api/HandleFiles/PostUploadFiles

Body Parameters

Data Type	Example
FormData	File: form_data.append('file', myFiles.File) With Folder: form_data.append('file', myFiles.File, FolderPath + "/" + myFiles.File.name)

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”
String	Destination	Destination alias (“Default” for the default vault).
String	Profile	Profile alias
String	FilePass	File password (in case the file is locked with a password)
String	ExtApp	External application name

Response

Data Type	Value
JSON/DataTable	[File Name, File Unique ID, Engine job ID]

JS Code Example

```
var form = new FormData();
form.append("file", fileInput.files[0], "test/test.pdf");

var settings = {
  "url": "https://<Base Address>/api/HandleFiles/PostUploadFiles",
  "method": "POST",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123",
    "Destination": "Default",
    "Profile": "Default_Alias",
    "FilePass": "",
    "ExtApp": "DefaultDomeApplication"
  },
  "processData": false,
  "mimeType": "multipart/form-data",
  "contentType": false,
  "data": form
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response: [["test.pdf","123e4567-e89b-12d3-a456-426614174000","89012345-e89b-12d3-a456-426614174111"]]
```

GetUploadFilesQueue

Description

Get the files in-process (waiting, in queue, in scan, saving, finished).

URL Format

https://<Base Address>/api/HandleFiles/GetUploadFilesQueue

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”

Response

Data Type	Value
JSON/DataTable	[CreatedDate, Destination, DestinationType, EndPoint, ExternalApplicationName, FileGUID, FileName, FileSize, FolderPath, JOBID, Link, OneDriveUser, ProfileID, ScanInfo, ScanStatus, UpdateDate]

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/HandleFiles/GetUploadFilesQueue",
  "method": "GET",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123"
  },
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response:
/*[{CreatedDate: "2021-03-16T09:16:52.83" Destination: "Default" DestinationType: null
EndPoint: "net.tcp://192.168.1.1:15810/GSDA_API"
ExternalApplicationName: "DefaultDomeApplication"
FileGUID: "123e4567-e89b-12d3-a456-426614174789" FileName: "test.pdf" FileSize: 1530018
FolderPath: "test" JOBID: "123e4567-e89b-12d3-a456-426614174456" Link: null
OneDriveUser: "ExFFY8vUDnCrNDZHxqnnRQ==" ProfileID: 1 ScanInfo: "KEYIPKEY127.0.0.1"
ScanStatus: "In Scan [28%]" UpdateDate: "2021-03-16T09:17:01.197"}]*/
```

PostClearFinishedFromQueue

Description

Clear all the finished files from the in-process table.

URL Format

https://<Base Address>/api/HandleFiles/PostClearFinishedFromQueue

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”

Response

Data Type	Value
JSON/String	“true”

JS Code Example

```
var settings = {
 "url": "https://<Base Address>/api/HandleFiles/PostClearFinishedFromQueue",
 "method": "POST",
 "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123"
  },
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response: true
```

GetFileList

Description

Get a list of files and folders from a specific destination/path.

URL Format

https://<Base Address>/api/HandleFiles/GetFileList/{secPeriod}/{limit}

URI Parameters

Data Type	Name	Description
Int	secPeriod	Get only files created in the last X seconds (0 for limitless).
Int	limit	Get top X files (0 for limitless).

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”
String	vault	Vault alias (“vault” for default).
String	folder	Sub folder name (empty for root).
String	searchKey	Search by file name (empty for all).

Response

Data Type	Value	Description
JSON/Struct	{ String[][] Files, Int FilesCount, Int FolderCount }	Folder: [folder name, created date, keygssdfolderkey]. File: [file name, created date, size, identifier, external application]

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/HandleFiles/GetFileList/0/0",
  "method": "GET",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123",
    "vault": "vault",
    "folder": "test",
    "searchKey": ""
  },
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response: ["test.pdf", "2021-03-16 09:17:16", "1530068", "123e4567-e89b-12d3-a456-426614174789", ""]
```

GetScanHistory

Description

Get all the information regarding the history of the scans in the system.

URL Format

https://<Base Address>/api/ScanHistory/GetScanHistory

Header Parameters

Data Type	Name	Exmple	Description
GUID	token	123e4567-e89b-12d3-a456-426614174000	
DateTime	`fromDate`	'YYYY-MM-DD HH:mm:ss’	
DateTime	`toDate`	'YYYY-MM-DD HH:mm:ss'	
Int	`limit`	10	
String	`searchKey`	“test.doc”	leave empty for all the results
String	`column`	“File Name”	leave empty for to search all `columns`

Response

Data Type	Value	Description
Json/DataTable[]	[Alias (destination), ClientHostName, ConcatErrorMessage, Saved, ConcatLogMessage, CreatedTimeStamp, DestinationFileType, DomainName, DurationSeconds, Endpoint, ExternalApplicationName, FileGuid, FileHASH, FileName, FileNewHASH, FileNewSizeBytes, FileSizeBytes, IPAddress, JOBID, OutputFullFilePath, Profileid, Results, Saved, ScanTime, SourceFileType, SourceTrueFileType, TimeInScan, UserName, isInfected]	Table – scan information Note: admin receives the information regarding all users, while normal user receives the information only regarding himself.
	[AVGTimeScan, TotalFiles, TotalsizeKB, cntClean, cntFixed, cntInfected, cntdeleted]	Table1 – summary

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/ScanHistory/GetScanHistory",
  "method": "GET",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123",
    "fromDate": "2021-03-16 09:00:00",
    "toDate": "2021-03-16 10:00:00",
    "limit": "10",
    "searchKey": "",
    "column": ""
  },
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response:
/*Table: Array(1) 0:
    [Alias: "Default" ClientHostName: "XXXX" ConcatErrorMessage: ""
    ConcatLogMessage: "Passed Description : <OK>     [OK]"
    CreatedTimeStamp: "Mar 16 2021  9:17AM" DestinationFileType: "" DomainName: "XXX"
    DurationSeconds: 0 Endpoint: "net.tcp://192.168.1.1:15810/GSDA_API"
    ExternalApplicationName: "DefaultDomeApplication"
    FileGuid: "123e4567-e89b-12d3-a456-426614174789"
    FileHASH: "XXXX" FileName: "test.pdf"  FileNewHASH: "" FileNewSizeBytes: 1530018
    FileSizeBytes: 1530018 IPAddress: "127.0.0.1"
    JOBID: "123e4567-e89b-12d3-a456-426614174456" OutputFullFilePath: "XXXX\XXXXXX.pdf"
    Profileid: 1 Results: "Ok"  Saved: true ScanTime: "2021-03-16T09:16:58"
    SourceFileType: "pdf" SourceTrueFileType: "pdf" TimeInScan: 0 UserName: "XXX"
    isInfected: false] ,
Table1: Array(1) 0:
    AVGTimeScan: 33 TotalFiles: 1 TotalsizeKB: 1494 cntClean: 1 cntFixed: 0
    cntInfected: 0 cntdeleted: 0]*/
```

PostDownloadFiles

Description

Download files/folder from the portal.

URL Format

https://<Base Address>/api/HandleFiles/PostDownloadFiles

Body Parameters

Data Type	Name	Example
GUID	File Identifier	1. "123e4567-e89b-12d3-a456-426614174000" 2. For folder, send the folder path + name. 3. For Zip with multiple files, separate the GUIDs/folders with *gssd*: "123e4567-e89b-12d3-a456-426614174000*gssd*123e4567-e89b-12d3-a456-426614174001*gssd* folder path + name"

Header Parameters

Data Type	Name	Description
GUID	token	"123e4567-e89b-12d3-a456-426614174000"
String	vault	Vault alias ("vault" for default).

Response

Data Type	Value
StreamContent	Requested file

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/HandleFiles/PostDownloadFiles",
  "method": "POST",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123",
    "vault": "vault",
    "Content-Type": "application/json"
  },
  "data": JSON.stringify("test/123e4567-e89b-12d3-a456-426614174789"),
};

$.ajax(settings).done(function (response) {
  var blob = new Blob([response], { type: "application/octet-stream" });
  var link = window.document.createElement('a');
  link.href = window.URL.createObjectURL(blob);
  link.download = "test.pdf";
  document.body.appendChild(link);
  link.click();
  URL.revokeObjectURL(link.href);
  document.body.removeChild(link);
});
```

PostDeleteFile

Description

Delete files from the vault.

URL Format

https://<Base Address>/api/HandleFiles/PostDeleteFile

Body Parameters

Data Type	Name	Example
GUID	File Identifier	1. “123e4567-e89b-12d3-a456-426614174000” 2. For multiple files, separate the GUIDs with *gssd*: “123e4567-e89b-12d3-a456-426614174000*gssd*123e4567-e89b-12d3-a456-426614174001”

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”
String	vault	Vault alias (“vault” for default).

Response

Data Type	Value
String	“true”

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/HandleFiles/PostDeleteFile",
  "method": "POST",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123",
    "vault": "vault",
    "Content-Type": "application/json"
  },
  "data": JSON.stringify("test/123e4567-e89b-12d3-a456-426614174789"),
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response: true
```

PostDeleteFolder

Description

Delete folders from the vault.

URL Format

https://<Base Address>/api/HandleFiles/PostDeleteFolder

Body Parameters

Data Type	Name	Example
String	Folder Identifier	1. “folder path + name” 2. For multiple folders, separate the names with *gssd*: “folder path + name*gssd*folder path + name”

Header Parameters

Data Type	Name	Description
GUID	token	“123e4567-e89b-12d3-a456-426614174000”
String	vault	Vault alias (“vault” for default).

Response

Data Type	Value
String	“true”

JS Code Example

```
var settings = {
  "url": "https://<Base Address>/api/HandleFiles/PostDeleteFolder",
  "method": "POST",
  "headers": {
    "token": "123e4567-e89b-12d3-a456-426614174123",
    "vault": "vault",
    "Content-Type": "application/json"
  },
  "data": JSON.stringify("test"),
};

$.ajax(settings).done(function (response) {
  console.log(JSON.parse(response));
});

//Response: true
```

Code Example (HTML & JS)

You can also find the example under [ROOT]/portal/GSSD_API_EXAMPLE.html.

```
<!DOCTYPE html>
<html>
<body>
    <!--<script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
    </script>-->
    <script src="../vendors/jquery/dist/jquery.min.js"></script>

    <div>Domain: <input type="text" id="inpDomain" /></div>
    <div>User: <input type="text" id="inpUser" /></div>
    <div>Password: <input type="text" id="inpPass" /></div>
    <div><input type="file" id="FileInput" /></div>

    <script>
        var BaseAddress = window.location.origin + "/api/";
        var async = false;
        var lastScanStatus;
        var HistoryFromDate = "2020-01-01 00:00:00";
        var HistoryToDate = "2030-01-01 00:00:00";

        $('#FileInput').change(function () {

            var domain = $('#inpDomain')[0].value;
            var user = $('#inpUser')[0].value;
            var password = $('#inpPass')[0].value;

            //START MAIN PROCESS
            console.log("Start: " + new Date());
            PrintHello();
            PrintGlobalValues();

            var ApiKey = PostReceiveExtAPIToken(domain, user, password);
            var SessionToken = PostExtLogin(ApiKey);
            PrintTokenActive(SessionToken);
            var ProfileAliasToUse = GetUserRules(SessionToken);
            PrintDestinations(SessionToken);
            console.log("Uploading File...");

            alert("START UPLOADING")
            PostUploadFiles(SessionToken, "Default",
                 ProfileAliasToUse, "", "DefaultDomeApplication");

            var FileGUIDandName = PrintUploadFilesQueueUntillDone(SessionToken);
            var FileGUID = FileGUIDandName.split(',')[0];
            var FileName = FileGUIDandName.split(',')[1];

            PostClearFinishedFromQueue(SessionToken);
```

```
        alert("START DOWNLOADING")

        PostDownloadFiles(SessionToken, "vault", FileGUID, FileName);
        PrintFileList(SessionToken, 0, 0, "vault", "", "");
        PrintScanHistory(SessionToken, HistoryFromDate, HistoryToDate,
              10, "", "");

        PostDeleteFile(SessionToken, "vault", FileGUID);
        PostLogOut(SessionToken);

        alert("FINISHED")
        console.log("Done: " + new Date());

    });

    //FUNCTIONS

    function PrintHello() {
        var settings = {
            "url": BaseAddress + "GSPortal/GetHello",
            "method": "GET",
            "async": async,
        };

        $.ajax(settings).done(function (response) {
            console.log(response);
        });
    }

    function PrintGlobalValues() {
        var settings = {
            "url": BaseAddress + "GSPortal/GetGlobalValues",
            "method": "GET",
            "async": async,
        };

        $.ajax(settings).done(function (response) {
            console.log("Global Settings:");
            console.log(JSON.parse(response));
        });
    }

    function PostReceiveExtAPIToken(domain, user, password) {
        var settings = {
            "url": BaseAddress + "GSPortal/PostReceiveExtAPIToken",
            "method": "POST",
            "async": async,
            "headers": {
                "domain": domain,
                "user": user,
                "captchaID": "",
                "captchaCode": "",
                "factorCode": "0",
                "ExtApp": "DefaultDomeApplication",
                "Content-Type": "application/json"
            },
            "data": JSON.stringify(password),
        };
```

```
            $.ajax(settings).done(function (response) {
                console.log("ApiKey: " + response);
                ApiKey = response;
            });

            return ApiKey;
        }

        function PostExtLogin(ApiKey) {
            var settings = {
                "url": BaseAddress + "GSPortal/PostExtLogin",
                "method": "POST",
                "async": async,
                "headers": {
                    "captchaID": "",
                    "captchaCode": "",
                    "factorCode": "0",
                    "ExtApp": "DefaultDomeApplication",
                    "Content-Type": "application/json"
                },
                "data": JSON.stringify(ApiKey),
            };

            var SessionToken;

            $.ajax(settings).done(function (response) {
                SessionToken = response.split(',')[0];
                console.log("Session Token: " + SessionToken);
            });

            return SessionToken;
        }

        function PrintTokenActive(SessionToken) {
            var settings = {
                "url": BaseAddress + "GSPortal/GetTokenActive",
                "method": "GET",
                "async": async,
                "headers": {
                    "token": SessionToken
                },
            };

            $.ajax(settings).done(function (response) {
                console.log("Session Token Is Active: " + response);
            });
        }

        function PostLogOut(SessionToken) {
            var settings = {
                "url": BaseAddress + "GSPortal/PostLogOut/false",
                "method": "POST",
                "async": async,
                "headers": {
                    "token": SessionToken
                },
            };

            $.ajax(settings).done(function (response) {
                console.log("Log Out: " + JSON.parse(response));
```

```
        });
    }

    function GetUserRules(SessionToken) {
        var settings = {
            "url": BaseAddress + "ApiRule/GetUserRules",
            "method": "GET",
            "async": async,
            "headers": {
                "token": SessionToken
            },
        };

        var ProfileAliasToUse;

        $.ajax(settings).done(function (response) {
            console.log("User Rules:");
            console.log(JSON.parse(response));
            ProfileAliasToUse = JSON.parse(response).profileAlias[0];
        });

        return ProfileAliasToUse;
    }
    function PrintDestinations(SessionToken) {
        var settings = {
            "url": BaseAddress + "Destinations/GetDestinations/false",
            "method": "GET",
            "async": async,
            "headers": {
                "token": SessionToken
            },
        };

        $.ajax(settings).done(function (response) {
            console.log("User Destinations:");
            console.log(JSON.parse(response));
        });
    }

    function PostUploadFiles(SessionToken, Destination, Profile, FilePass,
                             ExtApp) {
        var form = new FormData();
        form.append("file", FileInput.files[0]);

        var settings = {
            "url": BaseAddress + "HandleFiles/PostUploadFiles",
            "method": "POST",
            "async": async,
            "headers": {
                "token": SessionToken,
                "Destination": Destination,
                "Profile": Profile,
                "FilePass": FilePass,
                "ExtApp": ExtApp
            },
            "processData": false,
            "mimeType": "multipart/form-data",
            "contentType": false,
            "data": form
        };
```

```
        $.ajax(settings).done(function (response) {
            console.log( JSON.parse(response));
        });
    }

    function PrintUploadFilesQueueUntillDone(SessionToken) {
        var settings = {
            "url": BaseAddress + "HandleFiles/GetUploadFilesQueue",
            "method": "GET",
            "async": async,
            "headers": {
                "token": SessionToken
            },
        };

        var FileGUIDandName;

        $.ajax(settings).done(function (response) {
            if (lastScanStatus != JSON.parse(response)[0].ScanStatus)
                console.log("Scan Status: " +
                     JSON.parse(response)[0].ScanStatus);

            lastScanStatus = JSON.parse(response)[0].ScanStatus;

            if (lastScanStatus.indexOf("Finished") == -1) {
                //delay
                var d1 = new Date(); var d2 = new Date();
                while (d2.valueOf() < d1.valueOf() + 3000) d2 = new Date();

                PrintUploadFilesQueueUntillDone(SessionToken);
            }

            FileGUIDandName = JSON.parse(response)[0].FileGUID + "," +
                 JSON.parse(response)[0].FileName;
        });

        return FileGUIDandName;
    }

    function PostClearFinishedFromQueue(SessionToken) {
        var settings = {
            "url": BaseAddress + "HandleFiles/PostClearFinishedFromQueue",
            "method": "POST",
            "async": async,
            "headers": {
                "token": SessionToken
            },
        };

        $.ajax(settings).done(function (response) {
            console.log("ClearFinishedFromQueue: " + JSON.parse(response));
        });
    }

    function PrintFileList(SessionToken, secPeriod, limit, vault, folder,
            searchKey) {
        var settings = {
            "url": BaseAddress + "HandleFiles/GetFileList/" + secPeriod + "/"
                    + limit,
```

```
                "method": "GET",
                "async": async,
                "headers": {
                    "token": SessionToken,
                    "vault": vault,
                    "folder": folder,
                    "searchKey": searchKey
                },
            };

            $.ajax(settings).done(function (response) {
                console.log("File List From The Vault:");
                console.log(JSON.parse(response));
            });
        }

        function PrintScanHistory(SessionToken, fromDate, toDate, limit,
                  searchKey, column) {
            var settings = {
                "url": BaseAddress + "ScanHistory/GetScanHistory",
                "method": "GET",
                "async": async,
                "headers": {
                    "token": SessionToken,
                    "fromDate": fromDate,
                    "toDate": toDate,
                    "limit": limit,
                    "searchKey": searchKey,
                    "column": column
                },
            };

            $.ajax(settings).done(function (response) {
                console.log("Scan History:");
                console.log(JSON.parse(response));
            });
        }

        function PostDownloadFiles(SessionToken, vault, FileGUID, FileName) {
            $.ajax({
                url: BaseAddress + "HandleFiles/PostDownloadFiles",
                cache: false,
                type: 'POST',
                headers: {
                    token: SessionToken,
                    vault: vault
                },
                data: JSON.stringify(FileGUID),
                contentType: "application/json",
                xhr: function () {
                    var xhr = new XMLHttpRequest();
                    xhr.open("POST", BaseAddress + "HandleFiles/PostDownloadFiles");
                    xhr.responseType = 'blob';
                    xhr.addEventListener("progress", function (evt) {
                        if (evt.lengthComputable) {
                            var percentComplete = ((evt.loaded / evt.total) * 100);
                            console.log("Download Progress: " +
parseInt(percentComplete) + "%");
                        }
                    }, false);
```

```
                    return xhr;
                },
                crossDomain: true,
                accept: "Application/octet-stream",
                responseType: "blob",
                async: true,
                success: function (response) {
                    var blob = new Blob([response], { type: "application/octet-stream"
});

                    if (window.navigator && window.navigator.msSaveOrOpenBlob) {
                        window.navigator.msSaveOrOpenBlob(blob, FileName);
                    }
                    else {
                        var link = window.document.createElement('a');
                        link.href = window.URL.createObjectURL(blob);
                        link.download = FileName;
                        document.body.appendChild(link);
                        link.click();
                        URL.revokeObjectURL(link.href);
                        document.body.removeChild(link);
                    }
                }
            });
        }

        function PostDeleteFile(SessionToken, vault, FileGUID) {
            var settings = {
                "url": BaseAddress + "HandleFiles/PostDeleteFile",
                "method": "POST",
                "async": async,
                "headers": {
                    "token": SessionToken,
                    "vault": vault,
                    "Content-Type": "application/json"
                },
                "data": JSON.stringify(FileGUID),
            };

            $.ajax(settings).done(function (response) {
                console.log("File Deleted: " + JSON.parse(response));
            });
        }
    </script>
</body>
</html>
```

END OF DOCUMENT