



Electronic Submission Gateway (ESG) NextGen

User Acceptance Test Guide for Industry Users:
Application Programming Interfaces

V3.0

November 20, 2024

Table of Contents

1	Purpose.....	2
2	ESG NextGen Introduction.....	2
3	UAT Introduction	3
3.1	UAT Schedule.....	3
3.1.1	Wave 1: July 2024 – August 2024.....	3
3.1.2	Wave 2: September 2024 - October 2024.....	3
3.1.3	Wave 3: October 2024 - November 2024.....	4
3.2	UAT Training and Support	4
4	Prerequisites to API Testing.....	4
4.1	Install the PowerShell App	4
4.1.1	From GitHub.....	4
4.1.2	Using the Windows Package Manager Winget Command-line Tool.....	5
5	API File Submission	6
5.1	Prepare Files	6
5.2	Obtain API Client Id and Secret.....	7
5.3	Update credentials.xml	9
5.4	Update the XML File Path.....	9
5.5	Execute the File Upload Script	10
5.6	Executing API File Upload Script.....	10
5.7	Check File Submission Status	14
	Appendix 1. Common Troubleshooting Issues.....	15
1.1	Versioning Mismatch	15
1.2	Multiple Versions of PowerShell	16
1.3	PowerShell Execution Policy	16
	Appendix 2. FDA_Center and Submission Types	18
	Appendix 3. API Specifications	20
	Appendix 4. API Upload Test Cases	57
	Appendix 5. Industry Partner APIs	58

1 Purpose

This User Acceptance Testing (UAT) guide is a supporting document for testers participating in the UAT of the Electronic Submission Gateway (ESG) Next Generation (ESG NextGen) Application Programming Interfaces (APIs). It introduces ESG NextGen and describes key ESG NextGen terms. This document, together with “UAT Testing Scripts for Industry Users, Unified Submission Portal (USP)” prepares and guides testers to conduct UAT for ESG NextGen.

2 ESG NextGen Introduction

ESG NextGen addresses some major pain points of the legacy ESG in user management, package preparation, and file submission. It also introduces new business processes and new technologies to streamline the submission process, enhance the user experience (UX) and improve security, as described below.

Submission Package Preparation: ESG NextGen modernizes the UX and replaces the legacy ESG User Interface (UI) with the new USP, which provides step-by-step guidance to prepare the submission package.

File Submission Improvements:

- ESG NextGen provides web service Application Programming Interfaces (APIs) to submit files and check submission status. The APIs enable companies to integrate FDA file submissions with their IT systems to achieve submission automation.
- ESG NextGen tracks the status of submissions via the USP, APIs, emails and FDA Acknowledgements.
- By leveraging modern cloud-based network technologies, ESG NextGen dramatically increases the file size upload capacity. In Release 1, The ESG NextGen can take up to a 1TB file submission. Further increases in file size will be developed in future ESG NextGen releases.

File Submission Options: ESG NextGen provides three interfaces for industry to submit files to FDA:

- **USP:** The USP was developed to address the business process integration, UX, self-service, and communication gaps of the legacy ESG. USP seamlessly integrates user registration, user account management, package preparation and submission, status tracking, reports and dashboards in a web-based UI with a modern look and feel, resulting in an a much-improved UX.
- **APIs:** ESG NextGen provides new web services APIs to submit files, check submission status and query other information. Companies can integrate these APIs with their systems to achieve submission and other business workflow automation.

- **Application Statement 2 (AS2):** ESG NextGen will continue to support AS2, an electronic data interchange protocol that is used in the legacy ESG to submit files to FDA.

3 UAT Introduction

ESG NextGen UAT is a critical phase in which participants are invited to test the end-to-end functionalities of ESG NextGen and provide feedback on system function the APIs, their UX, performance and anything else on which they care to comment.

3.1 UAT Schedule

The UAT will unfold in two waves, each targeting different functionalities within ESG NextGen. This manual outlines the testing schedule for each wave, and the API-related tasks in each wave.

3.1.1 Wave 1: July 2024 – August 2024

In Wave 1, all users will need to set up new accounts in ESG NextGen to test the user registration process. See the step-by-step guide to registering new users in Module [ESG NextGen UAT Training video](#).

Wave 1 tasks include:

- Test new user registration and user management features
- Test submissions via the USP
- Test submissions via APIs

3.1.2 Wave 2: September 2024 - October 2024

In Wave 2, user accounts will be migrated from the legacy ESG to ESG NextGen. Testers who have accounts in the legacy ESG will receive an email notification with the additional account registration steps required to activate their accounts in ESG NextGen.

Wave 2 tasks include:

- Test submissions via AS2
- Test existing ESG user accounts migration
- Re-test improvements from UAT feedback in Wave 1

3.1.3 Wave 3: October 2024 - November 2024

In Wave 3, legacy ESG data will be re-migrated into ESG NextGen.

Wave 3 tasks include:

- Test submissions via AS2, API, and/or USP
- Test existing ESG user accounts migration
- Review/test What's New items
- End to end testing with Centers

3.2 UAT Training and Support

Frequent Asked Questions (FAQs): UAT testers can check [ESG NextGen Frequently Asked Questions | FDA](#) for all FAQs related to ESG NextGen.

UAT Training: Demo videos, previous industry demo recordings, this UAT guide and UAT Testing Scripts are available at the [ESG NextGen UAT Landing Page](#). The ESG NextGen team plans to hold UAT kickoff and training sessions before UAT starts.

UAT Feedback Sessions: During both the Wave 1 and Wave 2 UAT phases, the ESG NextGen team plans to host weekly sessions to demonstrate ESG NextGen functions, answer questions, and collect feedback from testers. A detailed schedule of the sessions will be announced during the UAT kickoff meeting.

Technical Support: UAT testers can email esgngSupport@fda.hhs.gov with questions or to request technical support. Please assist the ESG NextGen Support Team in diagnosing your issue by including in your email a detailed description of the issue, any error messages, and screenshots if applicable.

4 Prerequisites to API Testing

The ESG NextGen team has developed executable scripts that integrate ESG NextGen APIs to obtain user and company information, submit files into the system, and check submission status. To run these scripts, the PowerShell application needs to be installed. Users can use any of the methods listed below to install the PowerShell App in their machine.

4.1 Install the PowerShell App

4.1.1 From GitHub

- a) Download the following PowerShell installation file from GitHub at <https://github.com/PowerShell/PowerShell/releases/download/v7.4.2/PowerShell-7.4.2-win-x64.msi>

- b) Double-click on the downloaded file and complete the PowerShell App installation process.

```
C:\Program Files\WindowsApps\Microsoft.PowerShell_7.4.0.0_x64_8wekyb3d8bbwe\pwsh.exe
PowerShell 7.4.0
PS C:\Windows\System32>
```

Open the command line and type “pwsh” to validate the PowerShell app version (7.4.0 or newer) is installed properly.

```
C:\Users\barun.Poudel\ pwsh <enter>
PowerShell 7.4.0
```

4.1.2 Using the Windows Package Manager Winget Command-line Tool

An alternative way to install PowerShell App is to use the Microsoft Windows winget package.

Notes:

- The Windows Package Manager winget command-line tool is available in Windows 11 and modern versions of Windows 10 as a part of the App Installer.
 - If Admin privileges are required to install the PowerShell app using a winget package, contact your IT Support for assistance.
- a) Type “cmd” in Windows Search and open the Command Prompt window.
- b) Type the following command to search for the latest version of PowerShell.

```
winget search Microsoft.PowerShell
```

The output from the command will look like the following:

Name	Id	Version	Source
PowerShell	Microsoft.PowerShell	7.4.1.0	winget
PowerShell Preview	Microsoft.PowerShell.Preview	7.4.0.101	winget

- c) Type the following command to install PowerShell or PowerShell Preview using the id parameter.

```
winget install --id Microsoft.Powershell --source winget
winget install --id Microsoft.Powershell.Preview --source winget
```

5 API File Submission

5.1 Prepare Files

- a) Download the ESGNG zip file (xml file, PowerShell scripts) from the [ESG NextGen UAT Landing Page](#) and extract the zip file to your local c:\ drive. This package includes all the files that are used for file submission via APIs.

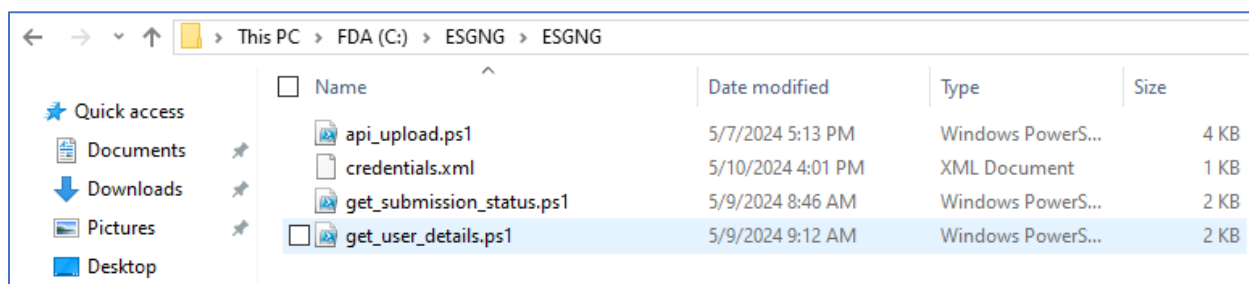


Figure 1: Copying all test files c:\ESGNG drive

- b) If you want to test different file formats, add or modify the file formats.
- c) If you want to save PowerShell scripts in a different location, you will need to modify the scripts because the XML files are configured with the original path. The API script should be modified so the path directory for the **api_Upload.ps1** script points to the modified path location.

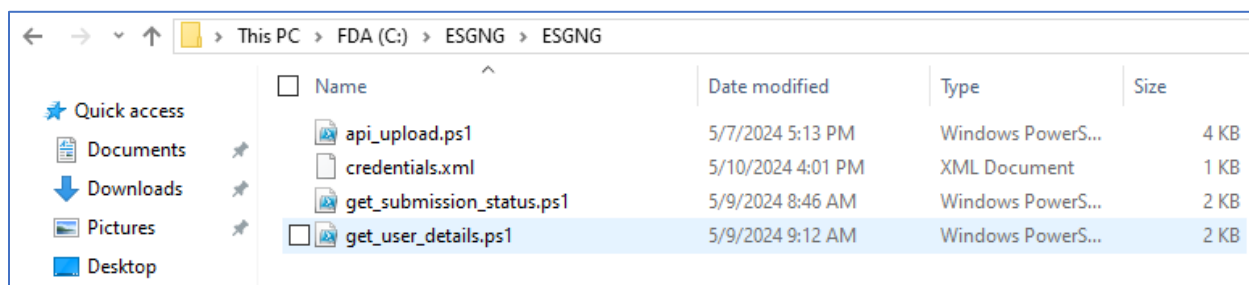


Figure 2. Copying all PowerShell script to desired location

5.2 Obtain API Client Id and Secret

To use ESG NextGen APIs, users are required to obtain Client Id and Secret for authentication. A Power User of a company registered in ESG NextGen has the privilege to obtain the Client Id and Secret, using the following steps:

1. Sign into the <https://appian-uat-dsc.fda.gov/> with the Power User email and password.

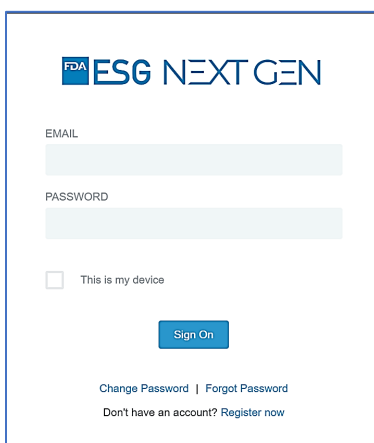
The screenshot shows the ESG Next Gen sign-in interface. At the top, there is a logo with 'FDA' in a blue box and 'ESG NEXT GEN' in blue text. Below the logo are two input fields: 'EMAIL' and 'PASSWORD'. Under the password field is a checkbox labeled 'This is my device'. A blue 'Sign On' button is centered below the checkbox. At the bottom, there are two links: 'Change Password | Forgot Password' and 'Don't have an account? Register now'.

Figure 3. Sign in to ESG Next USP

Check the Power User's email for the One Time Passcode (OTP) and enter it into the popup.

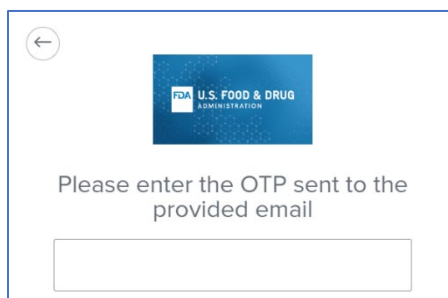
The screenshot shows a popup window for entering a One Time Passcode (OTP). It has a back arrow in the top left corner. In the center is the FDA U.S. Food & Drug Administration logo. Below the logo, the text reads 'Please enter the OTP sent to the provided email'. At the bottom is a single-line text input field.

Figure 4. Enter OTP

Login to the ESG NextGen USP main page.

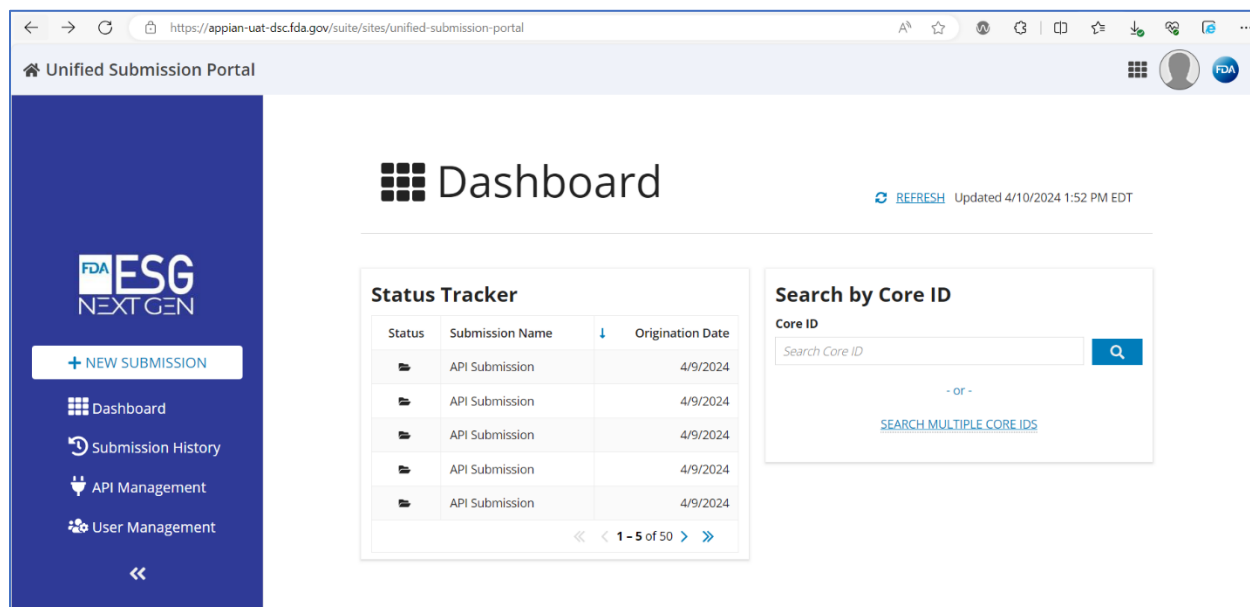


Figure 5. ESG NextGen home page

- d) Power Users have access to the API Management function. Click **API Management** in the left navigation bar
- e) Click **Generate New** to create a new set of Client Id and Secret.

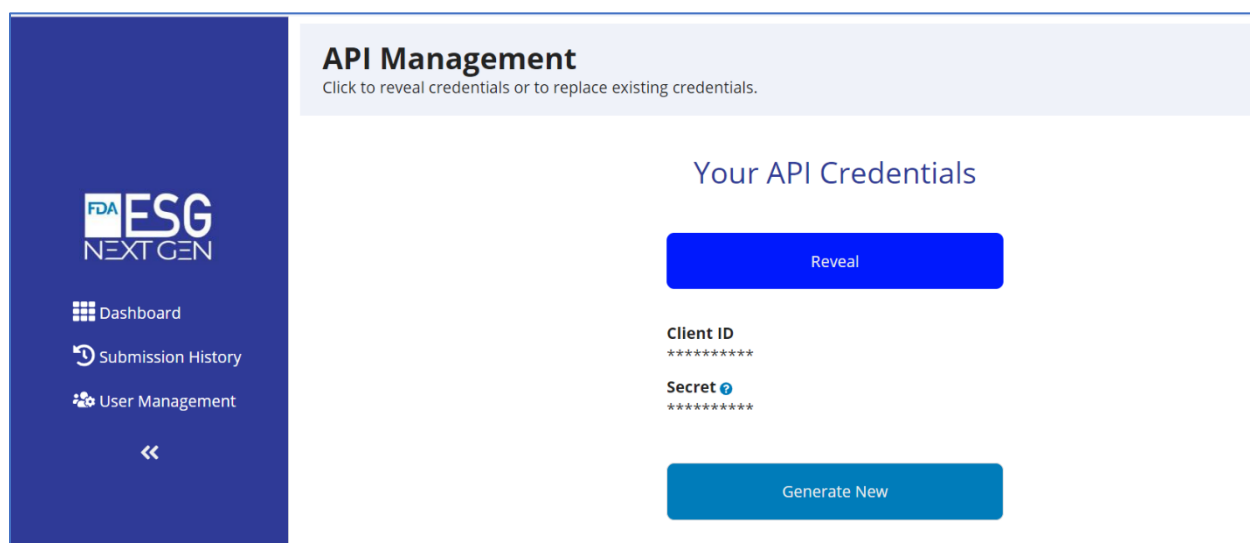


Figure 6: API Management Details

5.3 Update credentials.xml

Update the credentials.xml file located on C:\ESGNG with the Client Id and Secret information obtained in the previous subsection.

```
<?xml version="1.0" encoding="utf-8"?>

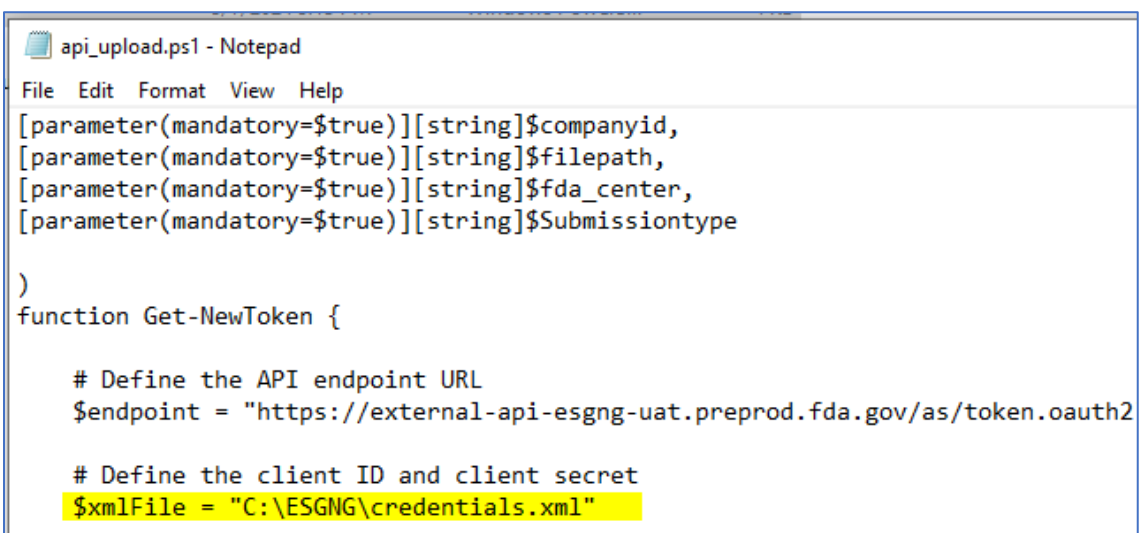
<credentials>
  <clientID>XX-XXXXXXXXXXXXXXXXXXXXXXXXXXXX</clientID>
  <clientSecret>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</clientSecret>
</credentials>
```

Figure 7. Update ClientID and Client Secret in credentials.xml

5.4 Update the XML File Path

Open the **get_user_details.ps1**, **api_upload.ps1**, **get_submission_status.ps1** scripts in PowerShell and update the correct path for the credentials.xml file.

Note: If the file path is not set properly, it will typically result in an error or an exception being raised, depending on the context in which the XML file is being assessed or used.



```
api_upload.ps1 - Notepad
File Edit Format View Help
[parameter(mandatory=$true)][string]$companyid,
[parameter(mandatory=$true)][string]$filepath,
[parameter(mandatory=$true)][string]$fda_center,
[parameter(mandatory=$true)][string]$Submissiontype
)
function Get-NewToken {
    # Define the API endpoint URL
    $endpoint = "https://external-api-esgng-uat.preprod.fda.gov/as/token.oauth2

    # Define the client ID and client secret
    $xmlFile = "C:\ESGNG\credentials.xml"
```

Figure 8. Update File Path for credentials.xml in Script

5.5 Execute the File Upload Script

- Go to the Windows search bar and type **PowerShell**.
- Navigate to the folder where the script is located.

```
cd c:\ESGNG
```

- Run the **pwsh** command to check PowerShell installation.

```
C:\ESGNG pwsh <enter>
```

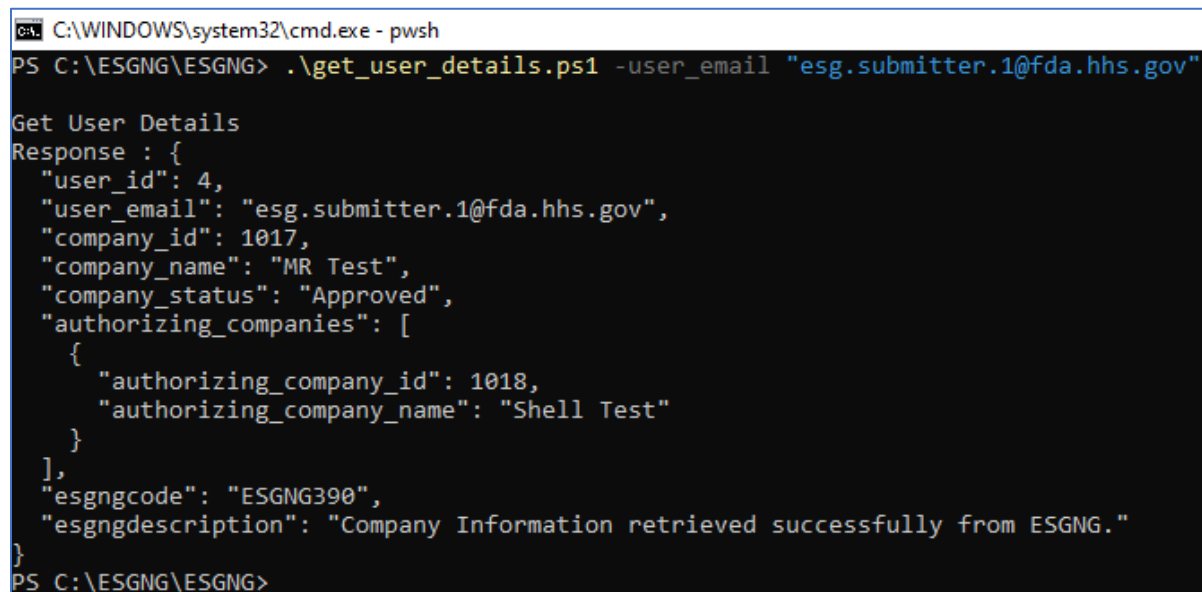
```
PowerShell 7.4.0
```

```
PS C:\ESGNG>
```

5.6 Executing API File Upload Script

- Run the PowerShell script to obtain the userid and companyid, two required file submission parameters. You will receive the response in an email.

```
PS C:\ESGNG > .\get_user_details.ps1 -user_email "xyz@gmail.com" <enter>
```



```
C:\WINDOWS\system32\cmd.exe - pwsh
PS C:\ESGNG\ESGNG> .\get_user_details.ps1 -user_email "esg.submitter.1@fda.hhs.gov"

Get User Details
Response : {
  "user_id": 4,
  "user_email": "esg.submitter.1@fda.hhs.gov",
  "company_id": 1017,
  "company_name": "MR Test",
  "company_status": "Approved",
  "authorizing_companies": [
    {
      "authorizing_company_id": 1018,
      "authorizing_company_name": "Shell Test"
    }
  ],
  "esgngcode": "ESGNG390",
  "esgngdescription": "Company Information retrieved successfully from ESGNG."
}
PS C:\ESGNG\ESGNG>
```

Figure 9. Response to Get User Details

If you wish to upload multiple files, combine all the files into a zip archive and then upload the zip file.

- b) In addition to the file path, the upload script requires additional parameters including FDA_Center and Submission Type for API upload. for [d:\ob\ob\Appendix 2. FDA Center and Submission Types\ob\](#) for details about both of those parameters.
- c) Run the PowerShell script to API upload. The script executes to create temp credentials for submission, initialize the submission, initialize the file upload, upload the file and finalize the file upload.

```
PS C:\ESGNG > .\api_upload.ps1 " <enter>
cmdlet api_upload.ps1 at command pipeline position 1
Supply values for the following parameters:
userid: 4
companyid: 1017
filepath: c:\esgng\testfiles\test.doc
fda_center: cber
Submissiontype: 510k
```

```
C:\WINDOWS\system32\cmd.exe - pwsh
PS C:\ESGNG\ESGNG> .\api_upload.ps1

cmdlet api_upload.ps1 at command pipeline position 1
Supply values for the following parameters:
userid: 4
companyid: 1017
filepath: c:\esgng\testfiles\test.doc
fda_center: cber
Submissiontype: 510k

Initialize submission and retrieve CoreID...
VERBOSE: Requested HTTP/1.1 POST with 223-byte payload
VERBOSE: Received HTTP/1.1 260-byte response of content type application/json
VERBOSE: Content encoding: utf-8

core_id           : MRTest_2024051027136
temp_user         : 4-MRTest_2024051027136
temp_password     : 08bedd26-fb57-42a9-bb9f-4d9648cc1520
esgngcode         : ESGNG210
esgngdescription  : Core Id and Temporary Credentials generated and stored successfully.

Initialize file upload...
05-10-2024 16:31:27
VERBOSE: Requested HTTP/1.1 GET with 0-byte payload
VERBOSE: Received HTTP/1.1 response of content type application/json of unknown size
VERBOSE: Content encoding: utf-8

Actually upload the file...
VERBOSE: Requested HTTP/1.1 POST with 1000215-byte payload
VERBOSE: Received HTTP/1.1 65-byte response of content type application/json
VERBOSE: Content encoding: utf-8

status : DRAFT
message : Draft saved successfully.

Days           : 0
Hours          : 0
Minutes        : 0
Seconds        : 0
Milliseconds   : 850
Ticks          : 8508452
TotalDays      : 9.84774537037037E-06
TotalHours     : 0.000236345888888889
TotalMinutes   : 0.0141807533333333
TotalSeconds   : 0.8508452
TotalMilliseconds : 850.8452

Finalize file upload...
VERBOSE: Requested HTTP/1.1 POST with 189-byte payload
VERBOSE: Received HTTP/1.1 33-byte response of content type application/json
VERBOSE: Content encoding: utf-8

message : Successful

PS C:\ESGNG\ESGNG> _
```

Figure 10. Output Response to File Upload

- d) Once the file upload is completed successfully, you will receive an Acknowledgement email (ACK 1).

From: donotreply@fda.hhs.gov <donotreply@fda.hhs.gov>
Sent: Thursday, March 7, 2024 11:49 AM
To: Karri, Nagendra Prakash * <NagendraPrakash.Karri@fda.hhs.gov>; ESGNG Support <esgngSupport@fda.hhs.gov>
Subject: ESG Submission - ACK1 - Upload Received: FantasticCalm_2024030737327

Submission FantasticCalm_2024030737327 has been successfully received by ESG.

Company Name: Boutique Bounty

Core ID: FantasticCalm_2024030737327

File Type: SINGLE

File Count Received: 1

File Count Identified to be Received: 1

Total File Size: 977 KB

Origination Date/Time: 2024-03-07T16:47:37

Next Step: Submission to be delivered to CBER

Figure 61. Acknowledgement 1 email: File Upload Received

- e) Once the file is submitted to a center, you will receive another Acknowledgement (ACK 2) email message confirming it was routed to the center.

From: donotreply@fda.hhs.gov <donotreply@fda.hhs.gov>
Sent: Thursday, March 7, 2024 11:49 AM
To: Karri, Nagendra Prakash * <NagendraPrakash.Karri@fda.hhs.gov>; ESGNG Support <esgngSupport@fda.hhs.gov>
Subject: ESG Submission - ACK2 - Submitted to Center FantasticCalm_2024030737327

Submission FantasticCalm_2024030737327 has been successfully submitted to CBER.

Company Name: Boutique Bounty

Core ID: FantasticCalm_2024030737327

File Type: SINGLE

File Count Received: 1

File Count Identified to be Received: 1

Total File Size: 977.0000 KB

Origination Date/Time: 2024-03-07T16:47:37

Next Step: CBER to review submission.

Figure 7. Acknowledgement 2 email: File Submitted to FDA Center

5.7 Check File Submission Status

Users can check the status of their submission request using the `get_submission_status` PowerShell script and the `core_id` for their request.

2. Obtain the `core_id` from Figure 8. [Output Response to File Upload](#) in the previous section.
3. Execute below PowerShell script.

```
PS C:\ESGNG> .\get_submission_status.1
cmdlet get_submission_status.ps1 at command pipeline position 1
Supply values for the following parameters:
core_id: MRTTest_2024041030357
```

Get Submission Status

Response: {

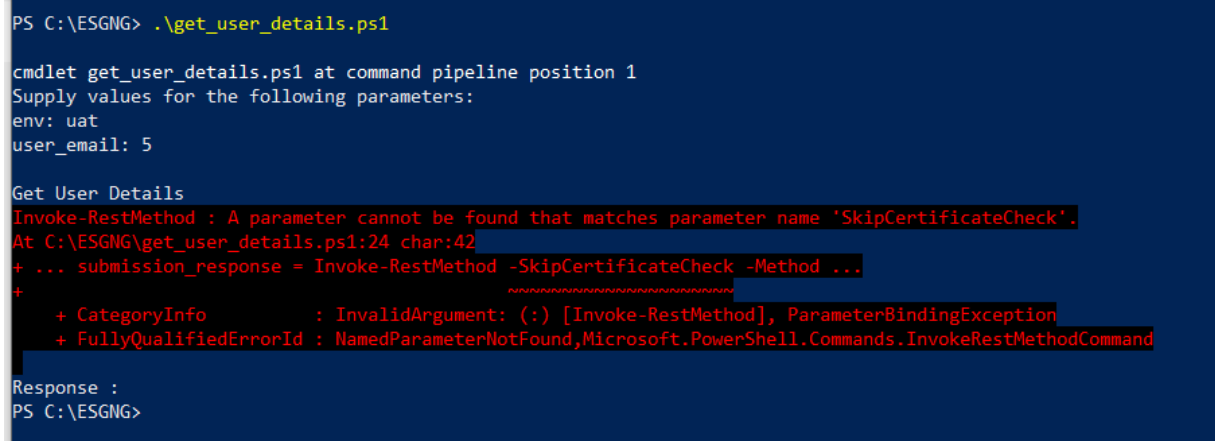
```
"core_id": "MRTTest_2024041030357",
"status": "Submitted to Center",
"submission_protocol": "API",
"esgngcode": "ESGNG270",
"esgngdescription": "Submission status retrieved successfully."
}
```

Appendix 1. Common Troubleshooting Issues

During the UAT, if you encounter a common issue described below, follow the recommended troubleshooting steps that may resolve it. You can also always reach out to the ESG NextGen Support Team for assistance by sending an email to esgngsupport@fda.hhs.gov.

1.1 Versioning Mismatch

The issue shown in Figure 11 is a script that is not compatible with the version of PowerShell being used (i.e., 7.4.X).



```
PS C:\ESGNG> .\get_user_details.ps1

cmdlet get_user_details.ps1 at command pipeline position 1
Supply values for the following parameters:
env: uat
user_email: 5

Get User Details
Invoke-RestMethod : A parameter cannot be found that matches parameter name 'SkipCertificateCheck'.
At C:\ESGNG\get_user_details.ps1:24 char:42
+ ... submission_response = Invoke-RestMethod -SkipCertificateCheck -Method ...
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Invoke-RestMethod], ParameterBindingException
+ FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.PowerShell.Commands.InvokeRestMethodCommand

Response :
PS C:\ESGNG>
```

Figure 83. Error on PowerShell scripts with Versioning Mismatch

- a) To check your PowerShell version:
 - i) Open PowerShell .
 - ii) Type **\$PSVersionTable.PSVersion** and press **Enter**
 - iii) If the version of PowerShell is not greater than 7.4.X, visit the official [PowerShell GitHub releases page](#) or the [Microsoft PowerShell website](#) to download and install the latest version.

1.2 Multiple Versions of PowerShell

The scripts will be executed only from the Windows PowerShell screen. The issue shown in Figure 14 is that there are two versions of PowerShell on the computer.

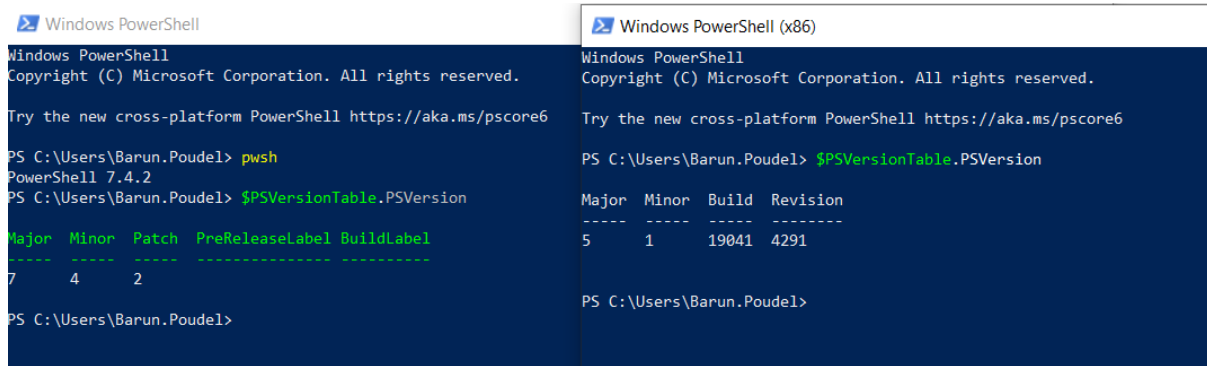


Figure 14. Different PowerShell Versions on the Same Machine

To resolve this issue:

- Delete the older version of PowerShell.
- If the remaining version of PowerShell is not greater than 7.4.X, visit the official [PowerShell GitHub releases page](#) or the [Microsoft PowerShell website](#) to download and install the latest version.

If you have trouble installing modules due to version mismatches, try using the - **RequiredVersion** parameter when installing the module to specify a compatible version.

- You can also use the **Install-Module -Scope CurrentUser** command if you encounter permission issues with installing modules systemwide.

1.3 PowerShell Execution Policy

If you encounter an issue after copying the files to the desired location, it may be because PowerShell's execution policy restricts the running of scripts for security reasons.

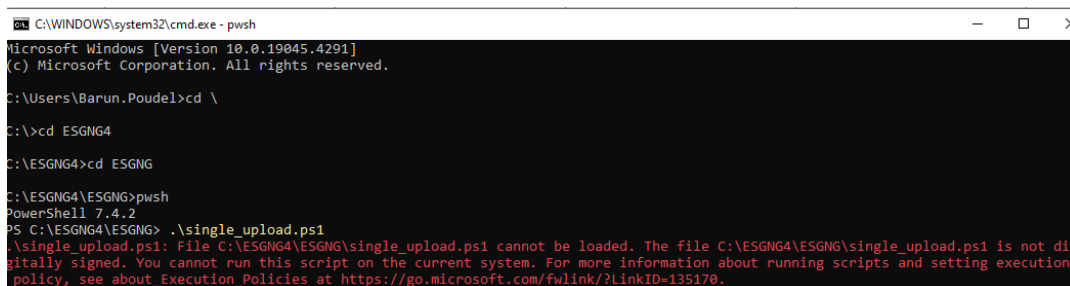


Figure 15. Execution Policy Error

To resolve this issue:

Navigate to the folder where the file is located and execute the following command.

```
PS C:\ESGNG> Set-ExecutionPolicy -ExecutionPolicy Unrestricted -  
Scope CurrentUser
```

If the above command does not fix the issue, execute the following commands from the folder where the files are located.

- a. Unblock-File get_submission_status.ps1
- b. Unblock-File api_upload.ps1
- c. Unblock-File get_user_details.ps1
- d. Unblock-File api_upload.ps1

Appendix 2. FDA_Center and Submission Types

FDA_Center	Submission Type	Additional Info
CDER	510K	
CDER	AERS	
CDER	AERS ATTACHMENTS	
CDER	AERS_ATTACHMENTS_PREMKT_CD	
CDER	AERS_PREMKT_CD	
CDER	BEST	
CDER	CDISC	
CDER	EBLA	
CDER	eDMF	
CDER	EIDE	
CDER	EIND	
CDER	EUA	
CDER	H1N1_Lot_Release	
CDER	Lot_Release_Protocol	
CDER	NDA	
CDER	PMA	
CDER	Pre_IND	
CDER	Promotional_Materials	
CDER	QSUBS	
CDER	SEND_PILOT	Test Only
CDER	SPL_LDD	
CDER	VAERS	
CDER	ACA6004_Drug_Samples	
CDER	AERS	
CDER	AERS Attachment IND	Test Only
CDER	AERS ATTACHMENTS	
CDER	AERS IND	Test Only
CDER	AERS_Attachments_PREMKT	Test Only
CDER	AERS_Attachments_PREMKT_CD	
CDER	AERS_PREMKT	Test Only
CDER	AERS_PREMKT_CD	
CDER	ECTD	
CDER	ECTD WAIVED	
CDER	EDMF TYPEIII	

CDER	EIND	
CDER	PFC	
CDER	FFU-PILOT	
CDER	GDUFA_Facility_Registration	
CDER	PFC	
CDER	POR Large Molecule Documents	
CDER	Voluntary_Direct_Aes	
CDRH	Adverse Events	
CDRH	Electronic_Submissions	
CDRH	GUDID	
CTP	Adverse_Events	
CTP	Electronic_Submissions	
CVM	Adverse_Events_Reports	
CVM	eSubmitter	
CVM	Electronic_Submissions	
CVM_VDD	eSubmitter	Test Only
HealthCanada	Transaction	
HFP	DSR_Adverse_Events	
HFP	EON-Payload-Files	
HFC	Food_Pilot_Listing	
HFC	Form3479	
HFC	Form3480	
HFC	Form3480A	
HFC	Form3503	
HFC	Form3665	
HFC	Form3666	
HFC	Form3667	
HFC	NDI	
HFC	Threshold_of_Regulation	
MWP	MWP_Report	Test Only
OC	OCAC	
OC	SPL	
OOPD	HUD_Designation_Requests	Test Only
OOPD	Orphan_drug_Designation_Requests	Test Only
ORA	Document_requests	
ORA	POR Small Molecule Documents	
OPQ	704a4_Pharma_Inspection_Records	

Appendix 3. API Specifications

API Name: OAuth2 Token Endpoint

Base URL: `https://external-api-esgng-uat.preprod.fda.gov`

Endpoint: `/as/token.oauth2`

HTTP Method: POST

Description:

This API provides an access token for client authentication using the OAuth2 `client_credentials` grant type. The token allows the client to authenticate and interact with other secured endpoints on the server.

Authentication:

- **Grant Type:** `client_credentials`
- **Scope:** `openid profile`

Request Details:

- **URL:**

`https://external-api-esgng-
uat.preprod.fda.gov/as/token.oauth2?grant_type=client_credentials&scope=openid%20profile`
- **Method:** POST
- **Headers:**
 - Content-Type: `application/x-www-form-urlencoded`
- **Body (form-data):**

`client_id=your_client_id
client_secret=your_client_secret`

```
curl --location 'https://external-api-esgng-
uat.preprod.fda.gov/as/token.oauth2?grant_type=client_credentials&scope=openid%20profile' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id= your_client_id' \
--data-urlencode 'client_secret= your_client_secret'
```

- **Status Code:** 200 OK
- **Response Body (JSON):**

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsIn...",
  "token_type": "Bearer",
  "expires_in": 3600,
}
```

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

```
HTTP POST on resource 'https://ssoext.preprod.fda.gov/as/token.oauth2'
failed: bad request (400).
```

- {

```
"esgngcode": "ESGNG403",  
  "esgngdescription": "Client ID is not associated with a valid  
user account.",  
  "message": "Bad request"  
}
```

- **Status Code:** 500 Internal Server Error

Response Output

```
{  
  "error": "Internal server_error",  
  "error_description": "An unexpected error occurred."  
}
```

Parameters:

Query Parameters:

- **grant_type:**
 - **Description:** The OAuth 2.0 grant type. In this case, it's `client_credentials`.
 - **Type:** string
 - **Required:** Yes
 - **Example:** `client_credentials`
- **scope:**
 - **Description:** Requested scopes of the access token. Here, the scopes are `openid` and `profile`.
 - **Type:** string
 - **Required:** Yes
 - **Example:** `openid profile`

Body Parameters (x-www-form-urlencoded):

- **client_id:**
 - **Description:** The client ID provided by the authorization server.
 - **Type:** string
 - **Required:** Yes
 - **Example:** `your_client_id`
- **client_secret:**
 - **Description:** The client secret corresponding to the client ID.
 - **Type:** string
 - **Required:** Yes

-
- **Example:** `your_client_secret`
-

Response Codes:

- **200 OK:** The access token was successfully generated.
- **400 Bad Request:** Invalid request, possibly due to missing parameters.
- **401 Unauthorized:** Authentication failure, incorrect client ID or secret.
- **500 Internal Server Error:** Unexpected server error.

API Name: ESGNG Credential Submission API PROD Submission

Base URL: `https://external-api-esgng-uat.preprod.fda.gov`

Endpoint: `/api/esgng/v1/credentials/api`

HTTP Method: POST

Description:

This API allows clients to submit credentials for a user to the FDA's ESGNG (Electronic Submissions Gateway Next Generation) system. The request requires an access token for authentication and includes various details about the submission.

Authentication:

- **Type:** Bearer Token (OAuth 2.0)
 - **Token:** The `access_token` obtained from the OAuth token endpoint.
-

Request Details:

- **URL:** `https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1/credentials/api`
- **Method:** POST
- **Headers:**
 - `Authorization: Bearer {access_token}`
 - `Content-Type: application/json`
- **Body (JSON):**

```
{  
  "user_id": "4505",  
  "fda_center": "CBER",  
  "company_id": "5842",  
  "submission_type": "510K",  
  "submission_name": "Test from Postman",  
  "submission_protocol": "API",  
  "file_count": 1,  
}
```

```
    "description": "this is submission version description."
  }
```

Request Example:

```
curl --location 'https://external-api-esgng-
uat.preprod.fda.gov/api/esgng/v1/credentials/api' \
--header 'access_token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9... ' \
--header 'Content-Type: application/json' \
--data '{
  "user_id": "4909",
  "fda_center": "CBER",
  "company_id": "5842",
  "submission_type": "510K",
  "submission_name": "Testing from Postman",
  "submission_protocol": "API",
  "file_count": 1,
  "description": "Testing from Postman"
}'
```

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{
  "core_id": "ci240911125314.86907a8a03df4427875feb2140a50bcc",
  "temp_user": "4909-ci240911125314.86907a8a03df4427875feb2140a50bcc",
  "temp_password": "eddbefca-a3d7-4b5a-805b-de0e68dd48ea",
  "esgngcode": "ESGNG210",
  "esgngdescription": "Core Id and Temporary Credentials generated and stored
successfully."
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{
  "esgngcode": "ESGNG219",
  "esgngdescription": "Invalid user, company, or center submission type. Please
check request inputs and try again.",
  "message": "Bad request"
}
```

- **Status Code:** 400 Invalid token
 - **Response Body (JSON):**

Response Output

```
{
  "detailedDescription": "HTTP POST on resource 'http://ESGSta-MyNLB-
L3p10ZLfCKpB-98584b6b2ed83b12.elb.us-gov-west-1.amazonaws.com/as/validate' failed:
internal server error (500).",
  "error": "Invalid Grant or Invalid Credentials"
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": "Internal server_error",
  "error_description": "An unexpected error occurred."
}
```

Parameters:

Header Parameters:

- **Authorization:**
 - **Description:** The Bearer token obtained via the OAuth 2.0 token endpoint.
 - **Type:** string
 - **Required:** Yes
 - **Example:** Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Body Parameters (JSON):

- **user_id:**
 - **Description:** The ID of the user making the submission which can be obtained by GetCompanyInfo API call
 - **Type:** string
 - **Required:** Yes
 - **Example:** "4505"
- **fda_center:**
 - **Description:** The FDA center associated with the submission (e.g., CBER, CDER, etc.).
 - **Type:** string
 - **Required:** Yes
 - **Example:** "CBER"
- **company_id:**
 - **Description:** The company ID associated with the submission which can be retrieved by GetCompanyInfo API call
 - **Type:** string
 - **Required:** Yes
 - **Example:** "5842"
- **submission_type:**
 - **Description:** The type of submission being made (e.g., 510K, NDA).
 - **Type:** string
 - **Required:** Yes
 - **Example:** "510K"
- **submission_name:**
 - **Description:** The name of the submission.
 - **Type:** string
 - **Required:** Yes
 - **Example:** "Test from Postman"
- **submission_protocol:**

- **Description:** The protocol used for the submission (e.g., API).
 - **Type:** `string`
 - **Required:** Yes
 - **Example:** "API"
 - **file_count:**
 - **Description:** The number of files included in the submission
 - **Type:** `integer`
 - **Required:** Yes
 - **Example:** 1
 - **description:**
 - **Description:** A description of the submission or submission version.
 - **Type:** `string`
 - **Required:** Yes
 - **Example:** "Submission details"
-

Response Codes:

- **200 OK:** The submission was successful, and a submission ID is returned.
- **400 Bad Request:** Invalid input or missing required parameters.
- **401 Unauthorized:** Invalid or expired access token.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG Credential Submission API Test Submission

Base URL: `https://external-api-esgng-uat.preprod.fda.gov`

Endpoint: `/api/esgng/v1/credentials/api/test`

HTTP Method: POST

Description:

This API allows clients to submit credentials for a user to the FDA's ESGNG (Electronic Submissions Gateway Next Generation) system. The request requires an access token for authentication and includes various details about the submission.

Authentication:

- **Type:** Bearer Token (OAuth 2.0)
 - **Token:** The `access_token` obtained from the OAuth token endpoint.
-

Request Details:

- **URL:** `https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1/credentials/api/test`
- **Method:** POST
- **Headers:**
 - `Authorization: Bearer {access_token}`
 - `Content-Type: application/json`
- **Body (JSON):**

```
{
  "user_id": "4505",
  "fda_center": "CBER",
  "company_id": "5842",
  "submission_type": "510K",
  "submission_name": "Test from Postman",
  "submission_protocol": "API",
  "file_count": 1,
```

```
    "description": "this is submission version description."
  }
```

Request Example:

```
curl --location 'https://external-api-esgng-
uat.preprod.fda.gov/api/esgng/v1/credentials/api/test' \
--header 'access_token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9... ' \
--header 'Content-Type: application/json' \
--data '{
  "user_id": "4909",
  "fda_center": "CBER",
  "company_id": "5842",
  "submission_type": "510K",
  "submission_name": "Testing from Postman",
  "submission_protocol": "API",
  "file_count": 1,
  "description": "Testing from Postman"
}'
```

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{
  "core_id": "ci240911125314.86907a8a03df4427875feb2140a50bcc",
  "temp_user": "4909-ci240911125314.86907a8a03df4427875feb2140a50bcc",
  "temp_password": "eddbefca-a3d7-4b5a-805b-de0e68dd48ea",
  "esgngcode": "ESGNG210",
  "esgngdescription": "Core Id and Temporary Credentials generated and stored
successfully."
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{
  "esgngcode": "ESGNG219",
  "esgngdescription": "Invalid user, company, or center submission type. Please
check request inputs and try again.",
  "message": "Bad request"
}
```

- **Status Code:** 400 Invalid token
 - **Response Body (JSON):**

Response Output

```
{
  "detailedDescription": "HTTP POST on resource 'http://ESGSta-MyNLB-
L3p10ZLfCKpB-98584b6b2ed83b12.elb.us-gov-west-1.amazonaws.com/as/validate' failed:
internal server error (500).",
  "error": "Invalid Grant or Invalid Credentials"
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": "Internal server_error",
  "error_description": "An unexpected error occurred."
}
```

Parameters:

Header Parameters:

- **Authorization:**
 - **Description:** The Bearer token obtained via the OAuth 2.0 token endpoint.
 - **Type:** string
 - **Required:** Yes
 - **Example:** Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Body Parameters (JSON):

- **user_id:**
 - **Description:** The ID of the user making the submission which can be obtained by GetCompanyInfo API call
 - **Type:** string
 - **Required:** Yes
 - **Example:** "4505"
- **fda_center:**
 - **Description:** The FDA center associated with the submission (e.g., CBER, CDER, etc.).
 - **Type:** string
 - **Required:** Yes
 - **Example:** "CBER"
- **company_id:**
 - **Description:** The company ID associated with the submission which can be retrieved by GetCompanyInfo API call
 - **Type:** string
 - **Required:** Yes
 - **Example:** "5842"
- **submission_type:**
 - **Description:** The type of submission being made (e.g., 510K, NDA).
 - **Type:** string
 - **Required:** Yes
 - **Example:** "510K"
- **submission_name:**
 - **Description:** The name of the submission.
 - **Type:** string
 - **Required:** Yes
 - **Example:** "Test from Postman"
- **submission_protocol:**

- **Description:** The protocol used for the submission (e.g., API).
 - **Type:** `string`
 - **Required:** Yes
 - **Example:** "API"
 - **file_count:**
 - **Description:** The number of files included in the submission
 - **Type:** `integer`
 - **Required:** Yes
 - **Example:** 1
 - **description:**
 - **Description:** A description of the submission or submission version.
 - **Type:** `string`
 - **Required:** Yes
 - **Example:** "Submission details"
-

Response Codes:

- **200 OK:** The submission was successful, and a submission ID is returned.
- **400 Bad Request:** Invalid input or missing required parameters.
- **401 Unauthorized:** Invalid or expired access token.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG File Payload (No Token Required)

Base URL: `https://upload-api-esgng.uat.preprod.fda.gov`

Endpoint: `/rest/forms/v1/fileupload/payload`

HTTP Method: GET

Description:

This API allows clients to retrieve file Payload ID or metadata without authentication via an access token. No input parameters are required for this request.

Authentication:

- **Type:** No authentication required for this request.

Request Details:

- **URL:** `https://upload-api-esgng-uat.preprod.fda.gov/rest/forms/v1/fileupload/payload`
- **Method:** GET
- **Headers:**
 - No authentication headers are required.
 - Content-Type: application/json

Request Example:

```
curl --location 'https://upload-api-esgng-uat.preprod.fda.gov/rest/forms/v1/fileupload/payload'
```

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{
  "data": {
    "payloadId": "6f10c466-b10a-43be-84d2-fc27cc661a28",
    "uploadFileLink": "https://upload-api-esgng-
uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/6f10c466-b10a-43be-84d2-
fc27cc661a28/file",
    "submitFormLink": "https://upload-api-esgng-
uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/6f10c466-b10a-43be-84d2-
fc27cc661a28/submit"
  }
}
```

Error Responses:

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": {
    "code": 500,
    "message": "Internal server error. Please contact the server
administrator.",
    "status": "INTERNAL_ERROR"
  }
}
```

Parameters:

- No input parameters are required for this request.
-

Response Codes:

- **200 OK:** File information successfully retrieved.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG File Upload API with Payload ID

Base URL: `https://upload-api-esgng-uat.preprod.fda.gov`

Endpoint: `/rest/forms/v1/fileupload/payload/{payloadID}/file`

HTTP Method: POST

Description:

This API allows clients to upload a file to the FDA's ESGNG system by sending a `multipart/form-data` request. The file is uploaded to a specific `payloadID` that is obtained from a previous request.

Authentication:

- **Type:** Bearer Token (OAuth 2.0)
- **Token:** The `access_token` obtained from the OAuth token endpoint.

Request Details:

- **URL:** `https://upload-api-esgng-uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/{payloadID}/file`
 - **Example:**
`https://upload-api-esgng-uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/5b5f4212-2fc1-4f63-b0e9-3ce62876dc00/file`
- **Method:** POST
- **Body (multipart/form-data):**
 - **file:** The binary file to be uploaded, specified by its file path.

Request Example:

```
curl --location 'https://upload-api-esgng-
uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/4c2bd633-160e-4374-bd32-
0f48d02b5132/file' \
--form 'file=@"/path/to/your/file.zip''
```

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{
  "data": {
    "status": "DRAFT",
    "message": "Draft saved successfully."
  }
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{
  "error": {
    "code": 400,
    "message": "Total number of upload files should be less than or equal to '1'.",
    "status": "INVALID_FILE_COUNT"
  }
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": {
    "code": 500,
    "message": "Internal server error. Please contact the server administrator.",
    "status": "INTERNAL_ERROR"
  }
}
```

Parameters:

URL Path Parameter:

- **payloadID:**
 - **Description:** The unique identifier for the payload from the previous request.
 - **Type:** string
 - **Required:** Yes
 - **Example:** 5b5f4212-2fc1-4f63-b0e9-3ce62876dc00

Header Parameters:

- **Authorization:**
 - **Description:** The Bearer token obtained via the OAuth 2.0 token endpoint.
 - **Type:** string
 - **Required:** Yes
 - **Example:** Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Body Parameters (multipart/form-data):

- **file:**
 - **Description:** The file to be uploaded.
 - **Type:** binary

- **Required:** Yes
 - **Example:** /path/to/your/file.zip
-

Response Codes:

- **200 OK:** File uploaded successfully, returns the file ID and payload ID.
- **400 Bad Request:** Invalid file format or missing fields.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG File Submit API

Base URL: `https://upload-api-esgng-uat.preprod.fda.gov`

Endpoint: `/rest/forms/v1/fileupload/payload/{payloadID}/submit`

HTTP Method: POST

Description:

This API allows clients to submit a payload by providing the `username`, `temp_password`, and `sha256_checksum` in the request body. No access token is required for this request.

Authentication:

- **Type:** No access token required for this request.

Request Details:

- **URL:**

`https://upload-api-esgng-
uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/{payloadID}/submit`

- **Example:**

`https://upload-api-
esgng.uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/19019
65f-f2a4-4e47-84a4-f6db16c07d00/submit`

- **Method:** POST

- **Headers:**

- Content-Type: application/json

- **Body (raw JSON):**

- `username`: The temp username associated with the request.
 - `password`: A temporary password for authentication.
 - `sha256_checksum`: The SHA-256 checksum of the payload to ensure data integrity.

Request Example:

```
curl --location 'https://upload-api-esgng-  
uat.preprod.fda.gov/rest/forms/v1/fileupload/payload/4c2bd633-160e-4374-bd32-  
0f48d02b5132/submit' \  
--header 'Content-Type: application/json' \  
--data '{  
  "username": "4909-ci240918122153.ec1012163cc944c2913d132e42ecd02a",  
  "password": "293dc314-219a-476a-a86e-6a64f174dfda",  
  "sha256_checksum":  
"B6630F7D0FF76F53F21B472F0D383B41A0B8A730A29282F12DB435DC390DFDEB"  
}'
```

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

Response Output

```
{  
  "data": {  
    "message": "Successful"  
  }  
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{
```

```
"data": {  
  "message": "Successful"  
}  
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{  
  "error": {  
    "code": 500,  
    "message": "Internal server error. Please contact the server  
administrator.",  
    "status": "INTERNAL_ERROR"  
  }  
}
```

Parameters:

URL Path Parameter:

- **payloadID:**
 - **Description:** The unique identifier for the payload to be submitted.
 - **Type:** string
 - **Required:** Yes
 - **Example:** 1901965f-f2a4-4e47-84a4-f6db16c07d00

Body Parameters (raw JSON):

- **username:**
 - **Description:** The unique username associated with the request.

- **Type:** string
- **Required:** Yes
- **Example:** "4847-ci240812190348.7beb65f0e2f747f5beb21effa03c34ab"
- **password:**
 - **Description:** A temporary password provided for authentication.
 - **Type:** string
 - **Required:** Yes
 - **Example:** "3420c46f-872f-4d4a-95f8-ed8639e1b653"
- **sha256_checksum:**
 - **Description:** The SHA-256 checksum of the file to verify integrity.
 - **Type:** string
 - **Required:** Yes
 - **Example:**
"d29751f2649b32ff572b5e0a9f541ea660a50f94ff0beedfb0b692b924cc8025"

Response Codes:

- **200 OK:** Payload successfully submitted, returns submission ID and payload ID.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG Get Submission Status API

Base URL: `https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1`

Endpoint: `/submissions/{core_id}`

HTTP Method: GET

Description:

This API retrieves the status of a specific submission in the ESGNG system by passing a `core_id` as a query parameter. The `core_id` is a unique identifier for the submission.

Authentication:

- **Type:** Bearer Token (OAuth 2.0)
- **Token:** The `access_token` obtained from the OAuth token endpoint.

Request Details:

- **URL:**

`https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1/submissions/core_id`

- **Example:**

`https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1/submissions/ci240911153205.2ce77ba34c8b4d668005518f2826386c`

- **Method:** GET
- **Headers:**
 - `Authorization: Bearer {access_token}`
 - `Content-Type: application/json`

Request Example:

```
curl --location --request GET 'https://external-api-esgng-  
uat.preprod.fda.gov/api/esgng/v1/submissions/ci240911153205.2ce77ba34c8b4d66  
8005518f2826386c' \  
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9... ' \  
--header 'Content-Type: application/json'
```

Query Parameters:

- **core_id:**
 - **Description:** The unique core ID for the submission you want to check the status of.
 - **Type:** string
 - **Required:** Yes
 - **Example:** ci240911153205.2ce77ba34c8b4d668005518f2826386c

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{  
  "core_id": "ci240911153205.2ce77ba34c8b4d668005518f2826386c",  
  "status": "Submitted to Center",  
  "submission_protocol": "USP",  
  "esgngcode": "ESGNG270",  
  "esgngdescription": "Submission status retrieved successfully."  
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{
  "esgngcode": "ESGNG271",
  "esgngdescription": "Invalid Request. Please check request inputs and try again.",
  "message": "Bad request"
}
```

- **Status Code:** Fail Response
 - **Response Body (JSON):**

Response Output

```
{
  "detailedDescription": "HTTP POST on resource 'http://ESGSta-MyNLB-L3p1OZLfCKpB-98584b6b2ed83b12.elb.us-gov-west-1.amazonaws.com/as/validate' failed: internal server error (500).",
  "error": null
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": {
    "code": 500,
```

```
    "message": "Internal server error. Please contact the server  
administrator.",  
  
    "status": "INTERNAL_ERROR"  
  }  
}
```

Response Codes:

- **200 OK:** Submission status retrieved successfully.
- **400 Bad Request:** The core ID is invalid or missing.
- **400 Unauthorized:** Invalid or expired access token.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG Get Companies by User Email API

Base URL: `https://external-api-esgng-uat.preprod.fda.gov`

Endpoint: `/api/esgng/v1/companies`

HTTP Method: GET

Description:

This API retrieves the list of companies associated with a specific user by passing the user's email address as a query parameter. The response provides details about the companies linked to the user.

Authentication:

- **Type:** Bearer Token (OAuth 2.0)
- **Token:** The `access_token` obtained from the OAuth token endpoint.

Request Details:

- **URL:** `https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1/companies?user_email={user_email}`
 - **Example:**
`https://external-api-esgng-uat.preprod.fda.gov/api/esgng/v1/companies?user_email=esg.submitter.1@fda.hhs.gov`
- **Method:** GET
- **Headers:**
 - `Authorization: Bearer {access_token}`
 - `Content-Type: application/json`

Request Example:

```
curl --location 'https://external-api-esgng-
uat.preprod.fda.gov/api/esgng/v1/submissions/ci240918122153.ec1012163cc944c2913d132e
42ecd02a' \
--header 'accesstoken: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...'
```

Query Parameters:

- **user_email:**
 - **Description:** The email address of the user whose associated companies need to be retrieved.
 - **Type:** string
 - **Required:** Yes
 - **Example:** esgngtest1@gmail.com

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{
  "user_id": 4909,
  "user_email": "esgngtest1@gmail.com",
  "company_id": 5842,
  "company_name": "Auth Shell myrtle",
  "company_status": "Approved",
  "esgngcode": "ESGNG390",
  "esgngdescription": "Company Information retrieved successfully from ESGNG."
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{
  "esgngcode": "ESGNG392",
  "esgngdescription": "User is invalid, inactive, or expired.",
  "message": "Bad request"
}
```

- **Status Code:** 400 Fail Response
 - **Response Body (JSON):**

Response Output

```
{
  "detailedDescription": "HTTP POST on resource 'http://ESGSta-MyNLB-L3p1OZLfCKpB-98584b6b2ed83b12.elb.us-gov-west-1.amazonaws.com/as/validate' failed: internal server error (500).",
  "error": "Invalid Grant or Invalid Credentials"
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": {
    "code": 500,
    "message": "Internal server error. Please contact the server administrator.",
    "status": "INTERNAL_ERROR"
  }
}
```

Response Codes:

- **200 OK:** Companies associated with the user were retrieved successfully.
- **400 Bad Request:** The email parameter is invalid or missing.
- **400 Fail Response:** Invalid or expired access token.
- **500 Internal Server Error:** Unexpected server-side error.

API Name: ESGNG Get Acknowledgement API

Base URL: `https://external-api-esgng-uat.preprod.fda.gov`

Endpoint: `/api/esgng/v1/acknowledgements/{acknowledgement_id}`

HTTP Method: GET

Description:

This API retrieves the details of a specific acknowledgment based on the acknowledgment ID provided in the URL path.

Authentication:

- **Type:** Bearer Token (OAuth 2.0)
- **Token:** The `access_token` obtained from the OAuth token endpoint.

Request Details:

- **URL:** `https://external-api-esgng-
uat.preprod.fda.gov/api/esgng/v1/acknowledgements/{acknowledgement_id}`
 - **Example:**
`https://external-api-
esgng.uat.preprod.fda.gov/api/esgng/v1/acknowledgements/ci240828
170030.0ce16ba6d3944975977c9594d0fabb74`
- **Method:** GET
- **Headers:**
 - `Authorization: Bearer {access_token}`
 - `Content-Type: application/json`

Request Example:

```
curl --location --request GET 'https://external-api-esgng-  
uat.preprod.fda.gov/api/esgng/v1/acknowledgements/ci240828170030.0ce16ba6d39  
44975977c9594d0fabb74' \  
  
--header 'access-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...'
```

Path Parameters:

- **acknowledgement_id:**
 - **Description:** The unique identifier for the acknowledgment you want to retrieve.
 - **Type:** string
 - **Required:** Yes
 - **Example:** ci240828170030.0ce16ba6d3944975977c9594d0fabb74

Response Details:

Successful Response:

- **Status Code:** 200 OK
- **Response Body (JSON):**

Response Output

```
{
  "core_id": "ci240828170030.0ce16ba6d3944975977c9594d0fabb74",
  "acknowledgements": [
    {
      "acknowledgement": "ACK1c",
      "status": "Upload Received",
      "message": "\n\n From: donotreply@fda.hhs.gov\n To:
esgngtest1@gmail.com\n Subject: ESG Submission - ACK1 - Upload Received:
ci240828170030.0ce16ba6d3944975977c9594d0fabb74\n\nSubmission
ci240828170030.0ce16ba6d3944975977c9594d0fabb74 has been successfully received by
ESG.\n\n Company Name: Auth Shell myrtle\n Core ID:
ci240828170030.0ce16ba6d3944975977c9594d0fabb74\n\tFile Type: SINGLE\n\tFile Count
Received: 1\n\tFile Count Identified to be Received: 1\n\tTotal File Size: 2.39
MB\n\tOrigination Date/Time: 08/28/2024 13:00:31 EDT\n\nNext Step: Submission to
be delivered to CTP",
      "created_by": "4909",
      "created_date": "08-28-2024, 17:06:43"
    },
    {
      "acknowledgement": "ACK2",
      "status": "Submitted to Center",
```

```
"message": "\n\n From: donotreply@fda.hhs.gov\n To:
esgngtest1@gmail.com\n Subject: ESG Submission - ACK2 - Submitted to Center:
ci240828170030.0ce16ba6d3944975977c9594d0fabb74\n\nSubmission
ci240828170030.0ce16ba6d3944975977c9594d0fabb74 has been successfully submitted to
CTP\n\n Company Name: Auth Shell myrtle\n\n\tCore ID:
ci240828170030.0ce16ba6d3944975977c9594d0fabb74\n\tFile Type: SINGLE\n\tFile Count
Received: 1\n\tFile Count Identified to be Received: 1\n\tTotal File Size: 2.39
MB\n\tOrigination Date/Time: 08/28/2024 13:00:31 EDT\n\tCenter Receipt Date/Time:
08/28/2024 13:06:43 EDT\n\t\tIf the center received your submission after 4:30 PM
EST, the official receipt date for the submission is the next government business
day.\n\nNext Step: CTP to review submission.",

    "created_by": "4909",

    "created_date": "08-28-2024, 17:06:44"

  }

],

  "esgngcode": "ESGNG360",

  "esgngdescription": "Retrieved all acknowledgements for the given submission."
}
```

Error Responses:

- **Status Code:** 400 Bad Request
 - **Response Body (JSON):**

Response Output

```
{

  "esgngcode": "ESGNG361",

  "esgngdescription": "Invalid Request. Please check request inputs and try
again.",

  "message": "Bad request"

}
```

- **Status Code:** Fail Response
 - **Response Body (JSON):**

Response Output

```
{
  "detailedDescription": "HTTP POST on resource 'http://ESGSta-MyNLB-
L3p1OZLfCKpB-98584b6b2ed83b12.elb.us-gov-west-1.amazonaws.com/as/validate' failed:
internal server error (500).",
  "error": null
}
```

- **Status Code:** 500 Internal Server Error
 - **Response Body:**

Response Output

```
{
  "error": {
    "code": 500,
    "message": "Internal server error. Please contact the server
administrator.",
    "status": "INTERNAL_ERROR"
  }
}
```

Response Codes:

- **200 OK:** Acknowledgment details retrieved successfully.
- **400 Bad Request:** The acknowledgment ID is invalid.
- **400 Unauthorized:** Invalid token.
- **500 Internal Server Error:** Unexpected server-side error.

Appendix 4. API Upload Test Cases

Test Case 10: Executing powershell Script for API File Upload				
Testing Objective: User leverages API upload functionality				
Prerequisites: Install Powershell App and and Download API script zip files from Landing Page				
<pre> graph LR A[Install Powershell App] --> B[Validate Powershell Installation] B --> C[Copy Powershell Script to C:\ Drive] C --> D[Login to ESGNG UI and generate Client Secrets] D --> E[Modify Credentials.XML file] E --> F[Execute Powershell Script] </pre>				
Step	Description	Expected Result	Pass/Fail	Actual Results (IF FAILED)
1	Install Powershell App	Verify the installation by opening a terminal or command prompt and typing pwsh		
2	Validate Powershell Installation on local machine	PowerShell prompt should appear, indicating that PowerShell is ready for use		
3	Download all required files for API Upload	Download the zip files from https://esgng-uat.preprod.fda.gov/portal/ to local machine and extract zip folder to the location where users are running the powershell script.		
4	Generate API credentials logging on to ESGNG UI	Login to ESGNG main screen select API Management and reveal ClientID and Client Secret and store them secure locally		
5	Update Credentials.XML file with client ID and ClientSecret	Credentials.XML file will be updated with user client ID and Client Secret Credentials		
7	Execute <code>get_user_details.ps1</code> powershell script	Execute powershell script with user_email parameter in powershell script. User details including user_id and Company_id will be generated as a response.		
8	Execute <code>api_upload.ps1</code> powershell script	Execute powershell script with user_id, Company_id, filepath, FDA_Center and Submission Type. The process will submit and file Initialize, Upload and Finalize the file upload. Users would receive acknowledgements (ACK 1 and ACK2) and will see all the upload details on the Submission history Tab on ESGNG User Interface.		
9	Execute <code>get_Submission_status.ps1</code> powershell script	Execute powershell script with core_id parameter in powershell script for the file upload Submission status. Users would get the reposonse of the file status according to the workflow (Upload Initiated, Upload in Progress, Upload Received, Upload Failed, Submitted to Center, Center's Response Sent)		
Secret key credentials can not be revealed more than once. If you lose the secret key and regenerated those details should be updated on Credentials.XML file.				

Appendix 5. Industry Partner APIs

No	API Endpoint	API Description (Industry Partners)	Request Method	Request Input (Sample values)	Request Output (Sample values)
1	https://external-api-esgng- uat.preprod.fda.gov /as/token.oauth2?g rant_type=client_cr edentials&scope=o penid%20profile	API to generate OAuth Token	POST	Request Body with x-www-form-urlencoded Keys -- Values: client_secret—Secret generated from API Management client_id -- ID generated from API Management NOTE: The values are constant which are from ICAM client credentials.	<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsImtpZCI6InNzb2tleSIsIng1dCI6In g4U3o5d1I1dmdjOHZZU0luLUM4cHUxRWVvUSIsInBp LmF0bSI6IjYtMWMifQ.eyJzY29wZSI6Im9wZW5pZCBwc m9maWxliwiYXV0aG9yaXphdGlzY29kZXRhaWxzIjpbXS wiY2xpZW50X2IkljoiZGNyLW11bGVzb2Z0IiwiaXNzIjoia HR0cHM6Ly9zc28uZmRhLmdvdilsmF1ZCI6IkkNUUF9IU 3VibWlzcjcvbl9Nb2RfREVWXX0V4dCIsImV4cCI6MTcwO TkYODkzM30.0R4m9O2q8Us3UOGnmQbv5J3vTYVKE_ kLhgvBfBmbco5gAAOtpLQYbkNIHtuZ8e9PHd6IMMtqD yD5i7klAdUE36z7UncVAp4zaND- xy2XixtOL5Gk0HQpwMMaBvJelWEhtUzoWWCnzvinU s2nu0Rwm0EK86sHlcXAO9KGBtOPsan_f4mhz82yhXGc rCqOlzwdHrXvwLB__kRkdj- xpCUAiTjYJ3txdncHL6frNP93QZ431kgGfKLPH3fzyvpF- KOHqTGSd0d2pDRmEc3DjsSlbU4pNT7B2Xe9BTOgVX_ Xt37_cHaziRqcb3zwitoUl3VvDAaZl2hNYnbuOikJZEhmy A", "token_type": "Bearer", "expires_in": 3600}</pre>
2	https://external-api- esgng- uat.preprod.fda.gov	Generate submission id and temp credentials	GET	Request Body: <pre>{ "user_id": "2",</pre>	<pre>{ "core_id": "CmpAssociatesInc_2023110733023", "temp_user": "1-CmpAssociatesInc_2023110733023",</pre>

No	API Endpoint	API Description (Industry Partners)	Request Method	Request Input (Sample values)	Request Output (Sample values)
	/api/esgng/v1/credentials/api			<pre>"fda_center": "CBER", "company_id": "2", "authorizing_company_id": "2", "submission_type": "510K", "submission_name": "Product Name7", "file_count": 23, "description": "this is submission version description", "submission_protocol": "API" }</pre> Note: Please pass accesstoken in a header Key – Values accesstoken - {accesstoken}	<pre>"temp_password": "17948854-108c-4831-9000-0859c52fefc0", "esgngcode": "ESGNG210", "esgngdescription": "Core Id and Temporary Credentials generated and stored successfully." }</pre>
3	https://external-api-esgng- uat.preprod.fda.gov/ api/esgng/v1/companies	Retrieve the user ID and list of companies and the associated company IDs for which a given user can create submissions	GET	Query Parameter: user_email = testagent5@test.com Note: Please pass accesstoken in a header Key – Values accesstoken - {accesstoken}	<pre>{ "user_id": 55, "user_email": "testagent5@test.com", "company_id": 30, "company_name": "Actual Agency", "company_status": "Approved", "authorizing_companies": [{ "authorizing_company_id": 2, "authorizing_company_name": "Oldio Corp" }, { "authorizing_company_id": 5, "authorizing_company_name": "Fantastic Calm" }]</pre>

No	API Endpoint	API Description (Industry Partners)	Request Method	Request Input (Sample values)	Request Output (Sample values)
					}, "esgngcode": "ESGNG390", "esgngdescription": "Company Information retrieved successfully from ESGNG." }
4	https://external-api-esgng- uat.preprod.fda.gov /api/esgng/v1/submissions/{core_id}	Retrieve submission status for a given core id	GET	URI Paramter: {core_id} -> CmpAssociatesInc_2023110733023 Note: Please pass accesstoken in a header Key – Values accesstoken - {accesstoken}	{ "core_id": "CmpAssociatesInc_2023110733023", "status": "Upload Initiated", "submission_protocol": "API", "esgngcode": "ESGNG270", "esgngdescription": "Submission status retrieved successfully." }
5	https://external-api-esgng- uat.preprod.fda.gov /api/esgng/v1/acknowledgements/{core_id}	Retrieve all ACKs information and also to have a link for ACK3 file to download.	GET	URI Paramter: {core_id} -> CmpAssociatesInc_2023110733023 Note: Please pass accesstoken in a header Key – Values accesstoken - {accesstoken}	
6	https://upload-api-esgng- uat.preprod.fda.gov /rest/forms/v1/fileupload/payload	Insert payload.	GET	Do not supply a request body with this method.	{ "data" : { "payloadId" : "string", "uploadFileLink" : "string", "submitFormLink" : "string"

No	API Endpoint	API Description (Industry Partners)	Request Method	Request Input (Sample values)	Request Output (Sample values)
					} }
7	https://upload-api-esgng- uat.preprod.fda.gov /rest/forms/v1/fileup load/payload/:paylo adId/file	Upload one attachment at a time.	POST	Request Parameter payloadId	{ "data" : { "status" : "string", "message" : "string" } }
8	https://upload-api-esgng- uat.preprod.fda.gov /rest/forms/v1/fileup load/payload/:Payl oadId/submit	Submit the payload.	POST	Request body: { "username" : "string", "password" : "string", "sha256_checksum" : "string" }	{ "data" : { "message" : "string", "files" : [{ "id" : "string", "name" : "string", "size" : "string", "link" : "string" }] } }