

Problem & Solution

Problem: when uploading file via a program it has a payload limit of 36 MB.

Solution: this limit can be overcome by uploading the file as multi-part/form data.

Step 1:

Upload the file to salesforce via postman, the file should be uploaded as a multi-part/form data.

Create new connected app

1. On setup search app manager
2. Click **New Connected App**

App Manager | Salesforce

Search Setup

Setup Home Object Manager

app manage

Apps

App Manager

Didn't find what you're looking for? Try using Global Search.

Clone Apps (Beta)

Quickly create new Lightning apps by cloning existing apps. To use the beta feature, indicate that you've read all legal requirements and agree to participate by toggling Enable App Cloning. See additional details and terms in the [Winter '23 release notes](#)

Enable App Cloning ☐ Disabled

34 items • Sorted by App Name • Filtered by All appmenuitems - TabSet Type

	App Name ↑	Developer Name	Description	Last Modified ...	App Type	Vi...
1	All Tabs	AllTabSet		9/21/2022, 4:28 PM	Classic	
2	Analytics Studio	Insights	Build CRM Analytics dashboards an...	9/21/2022, 4:28 PM	Classic	✓
3	Ant Migration Tool	Forcecom_Migration_Tool	The Force.com Migration Tool is a J...	9/21/2022, 4:28 PM	Connected (Managed)	
4	App Launcher	AppLauncher	App Launcher tabs	9/21/2022, 4:28 PM	Classic	✓
5	Bolt Solutions	LightningBolt	Discover and manage business solu...	9/21/2022, 4:31 PM	Lightning	✓

3. Fill in the details:

Connected App Name,

API Name,

Contact Email,

Enable the OAuth settings checkbox,

In callback URL put the link <https://login.salesforce.com>

In selected OAuth scopes, select **full access**

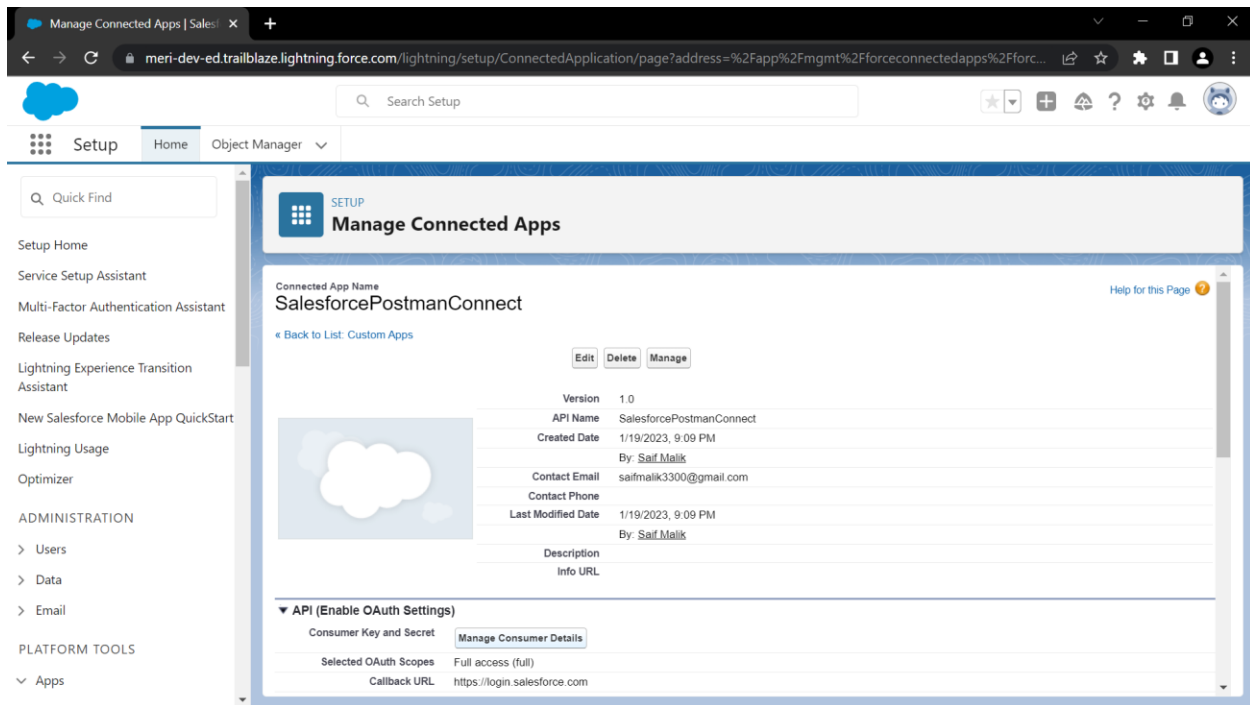
And, then **save**

The screenshot shows the Salesforce App Manager interface in a web browser. The browser's address bar displays the URL: `meri-dev-ed.trailblaze.lightning.force.com/lightning/setup/NavigationMenus/page?address=%2Fapp%2Fmgmt%2Fforceconnectedapps%2Fforc...`. The Salesforce navigation bar at the top includes the 'Setup' menu, a search bar, and various utility icons. On the left sidebar, the 'App Manager' link is highlighted under the 'Apps' section. The main content area is titled 'New Connected App' and contains a form with the following fields: 'Connected App Name', 'API Name', 'Contact Email', 'Contact Phone', 'Logo Image URL' (with a link to 'Upload logo image or Choose one of our sample logos'), 'Icon URL' (with a link to 'Choose one of our sample logos'), 'Info URL', and 'Description'. A legend indicates that red bars next to the first three fields denote 'Required Information'. At the bottom of the form, there is a section for 'API (Enable OAuth Settings)' with an 'Enable OAuth Settings' checkbox.

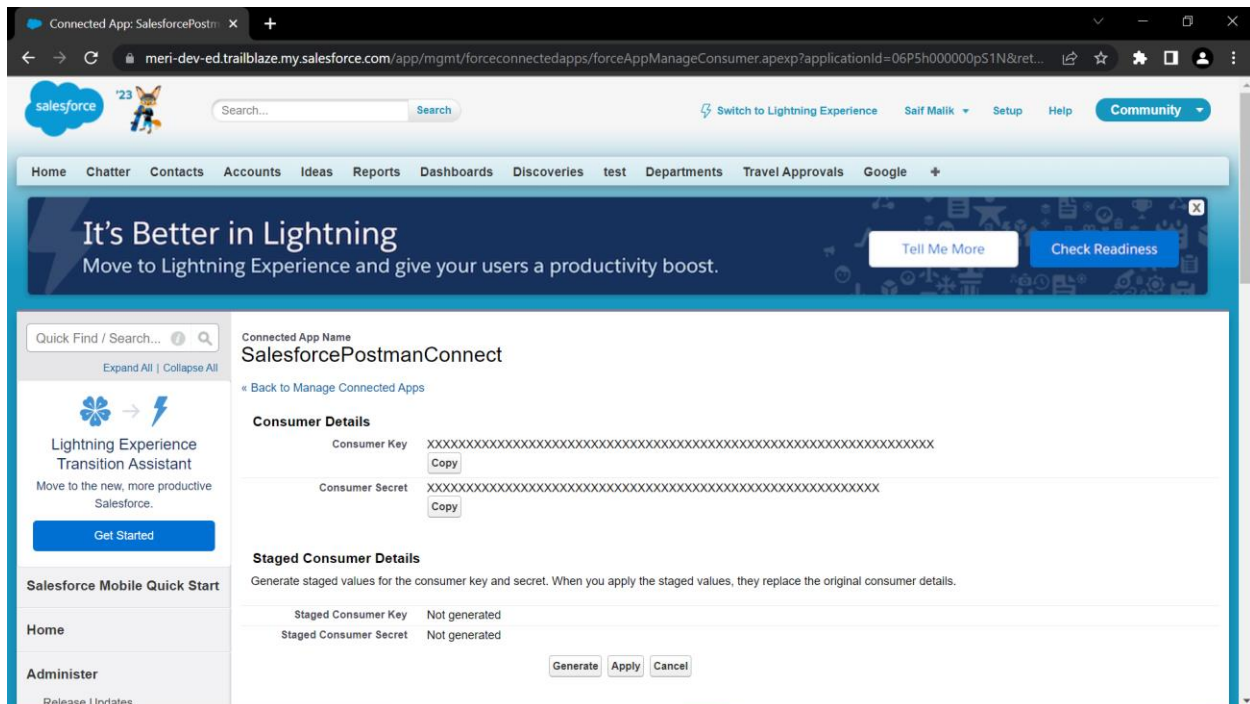
4. After the connected app is created, click **Manage Customer Details** to get client id & client secret

Get Client Id & Client Secret

1. After the connected app is created,
2. Click manage customer details to get **client id & client secret**
3. **Customer key is Client Id & Customer Secret is Client Secret**



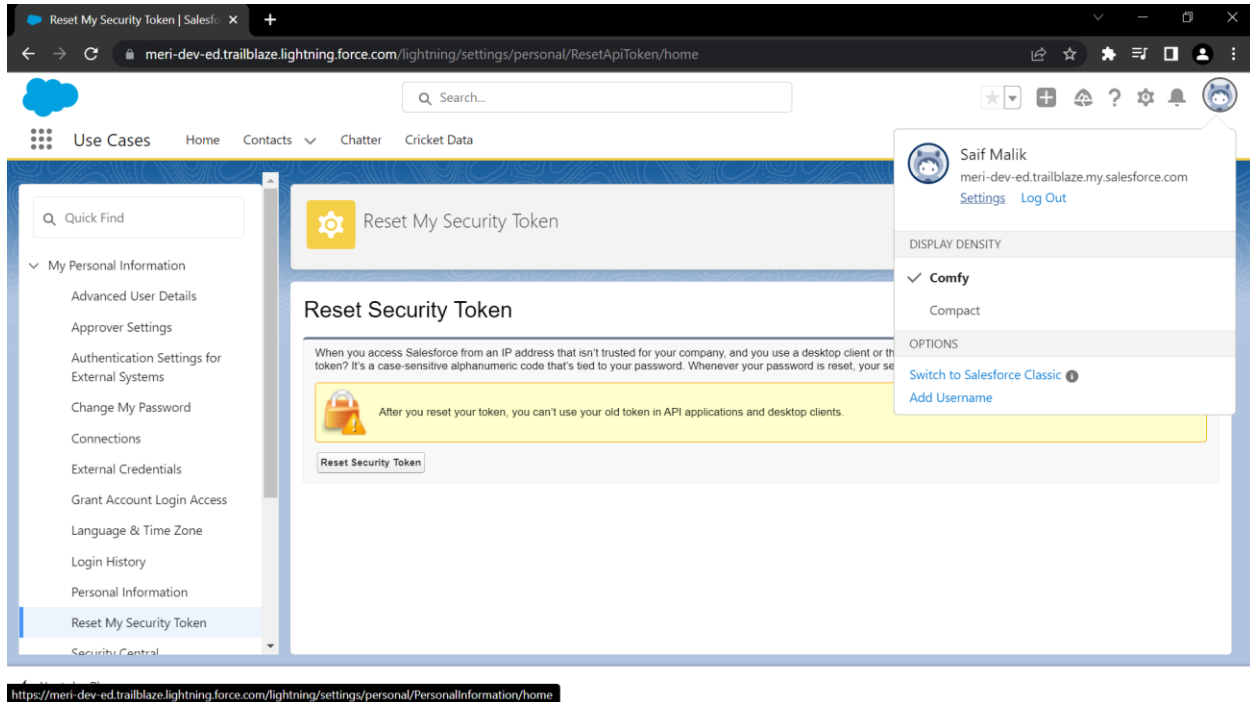
4. After **Manage Customer Details** is clicked this page will open.



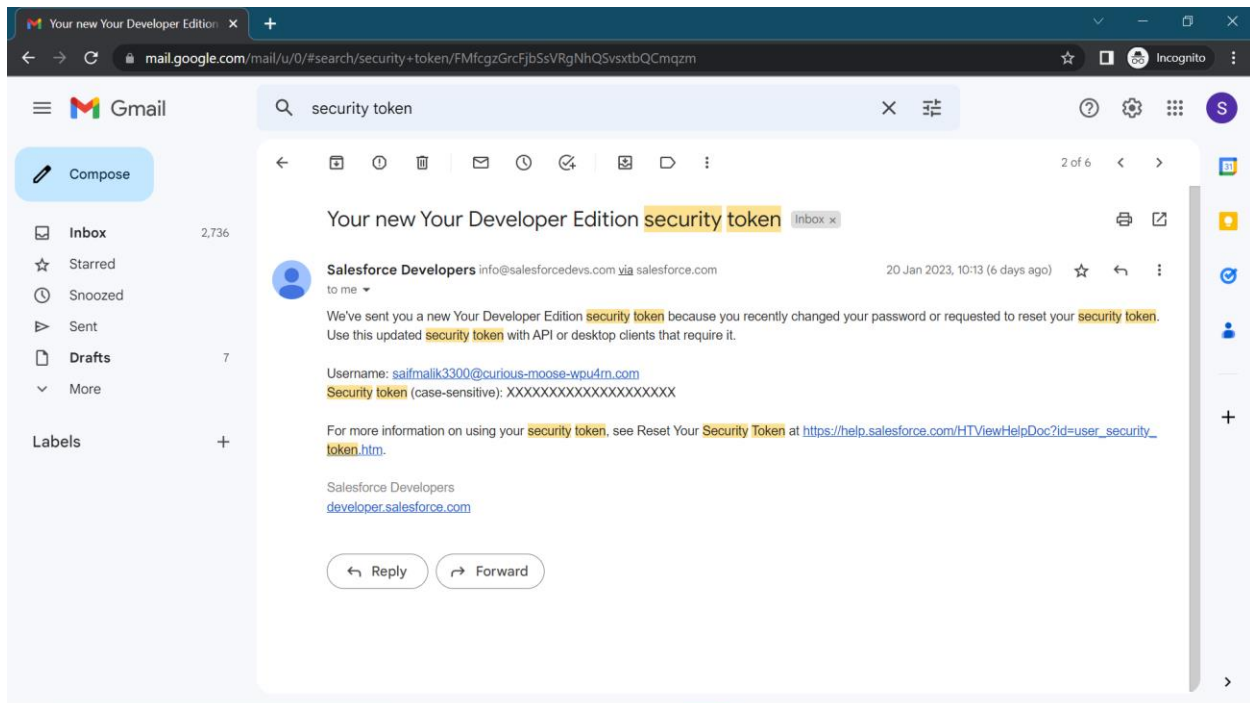
5. Copy **Customer Key** & **Customer Secret**

Get security token

1. Click view profile
2. Click settings
3. Under my personal information, open **Reset My Security Token**



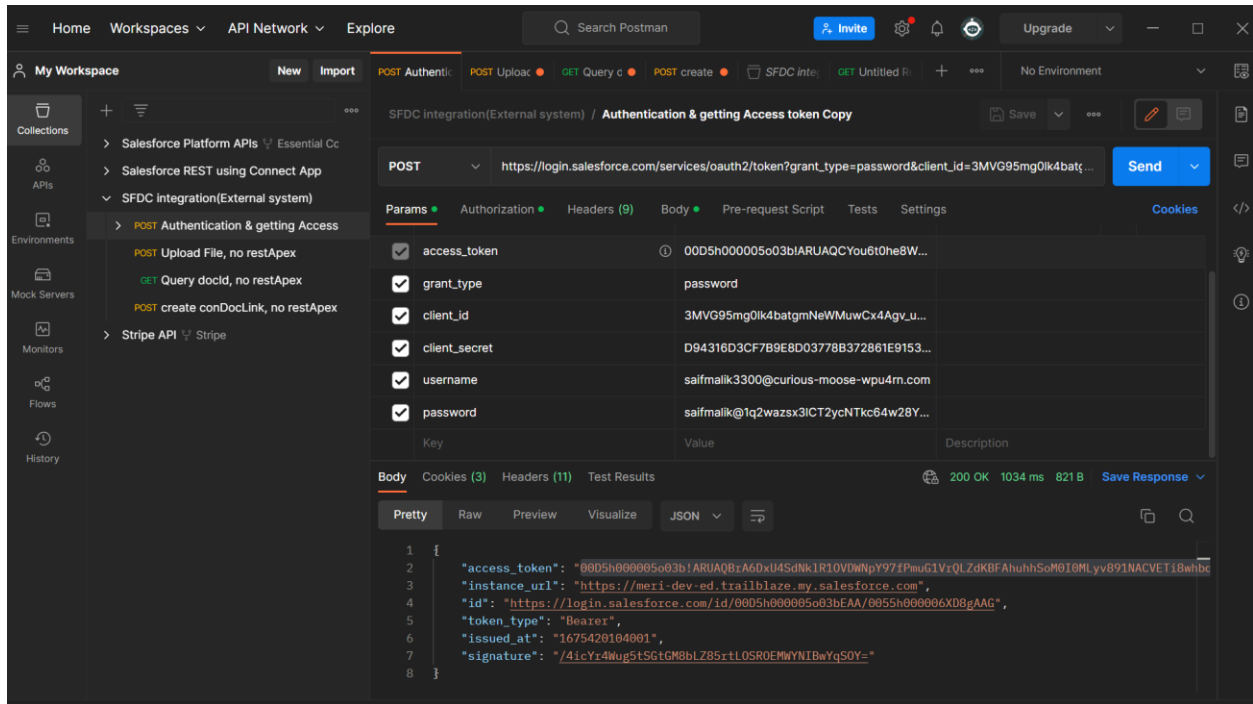
4. Then **Click Reset Security Token** button, this will send a mail with security token in it.



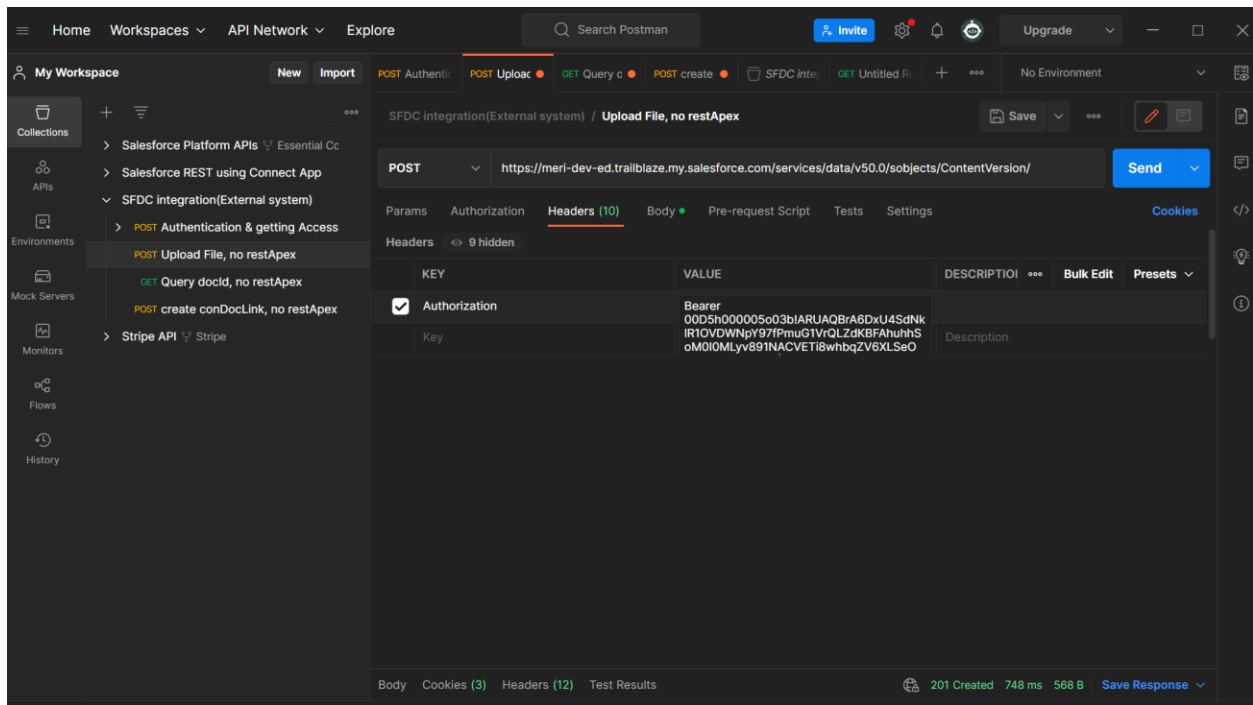
5. Since we have got the **Client Id**, **Client Secret** and **Secret token**.
6. get **username & password** handy
7. now I have got all the credentials, I can authenticate external system & send https requests.

Authentication (before sending requests to SFDC):

1. I will login to salesforce from external system, and I will get the access token so that I won't have to login again & again.
2. For logging in I will send a POST request to this(https://login.salesforce.com/services/oauth2/token?grant_type=password&client_id=<clientID>&client_secret=<clientSecret>&username=<username>&password=<password>&securityToken=<securityToken>) URL.



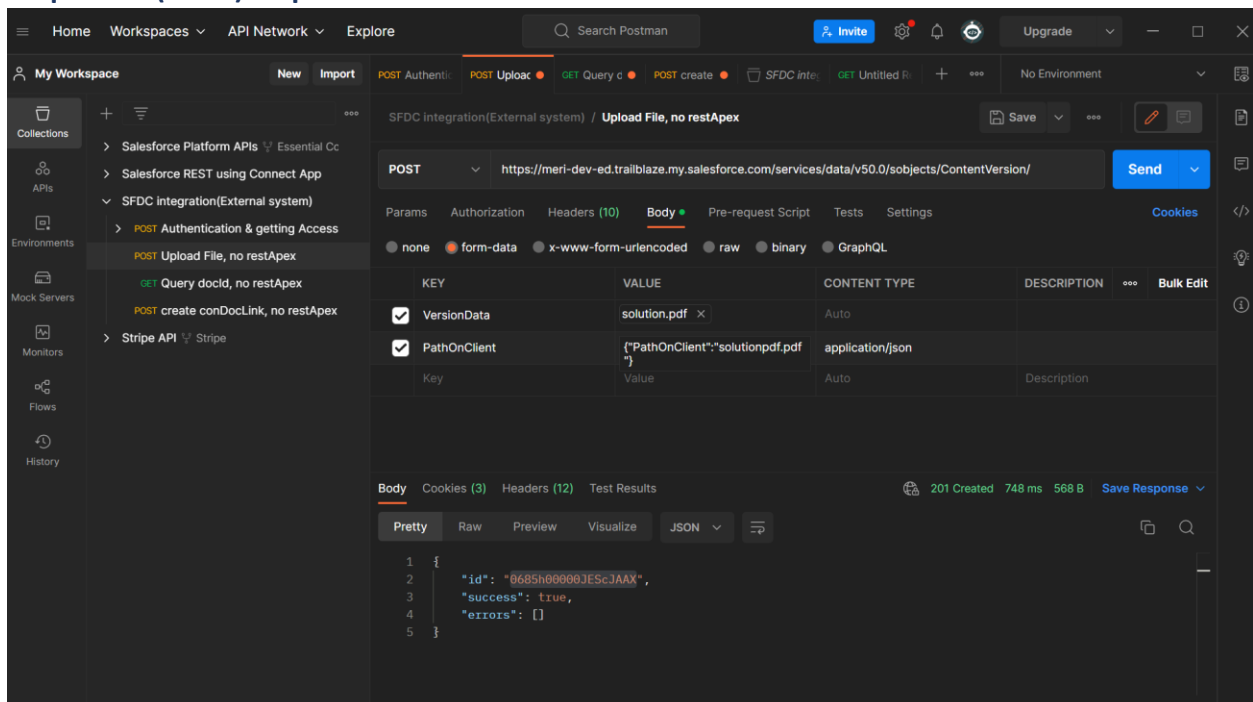
3. Now I have got the access token I will add it to authentication header of every request that will be send to this salesforce org.



4. Now we can send https requests to our salesforce org.

Solution:

Request 1: (POST): “upload file”



Files | Salesforce

meri-dev-ed.trailblaze.lightning.force.com/lightning/o/ContentDocument/home

Search...

Use CasesHomeContactsChatterCricket Data

Files

Owned by Me

4 items • Sorted by Last Modified Date

Upload Files

	Title	Owner	Last Modified Date ↓	
Owned by Me				
Shared with Me	solutionpdf	Saif Malik	2/3/2023, 4:03 PM	⌵
Recent	travel Asset File	Saif Malik	10/29/2022, 12:34 PM	⌵
Following	image Asset File	Saif Malik	9/23/2022, 6:46 PM	⌵
Libraries	thflogo Asset File	Saif Malik	9/21/2022, 4:28 PM	⌵

⚡ Youtube Player

Request 2: (GET): “query ContentDocumentId”

solutionpdf | Salesforce

meri-dev-ed.trailblaze.lightning.force.com/lightning/r/ContentDocument/0695h00000JAKrzAAH/view

Search...

Use CasesHomeContactsChatterCricket Data

File
solutionpdf

[+ Follow](#)[Download](#)[Share](#)[Public Link](#)

Size	File Extension	Owner
556KB	pdf	Saif Malik

PreviewDetails

Solution:


I am using the 3 HTTP requests sent via Postman:

through the first request i.e., a POST request, I am uploading the file

through the second request i.e., a GET request, I query the ContentDocumentId of the file uploaded in the previous request.

through a third request i.e., a POST request, I have created the ContentDocumentLink to link the file with a record.

Request 2: (GET) "query ContentDoc"



File Engagement ⓘ

Views

1

Downloads

0

Shared with (2)

Saif Malik
Owner

Teddy Kimbler

⚡ Youtube Player

Postman interface showing a GET request to the Salesforce API endpoint: `https://meri-dev-ed.trailblaze.my.salesforce.com/services/data/v50.0/query?q=SELECT Id, ContentDocu...`. The request is part of a collection named "SFDC Integration(External system)".

The request parameters are defined in the "Query Params" section:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> q	SELECT Id, ContentDocumentId, title FROM contentversion WHERE Id = '0685h00000JEScJAAX'	
Key		Description

The response is shown in the "Body" section, displaying a JSON object:

```
1 {
2   "type": "ContentVersion",
3   "url": "/services/data/v50.0/objects/ContentVersion/0685h00000JEScJAAX",
4   "Id": "0685h00000JEScJAAX",
5   "ContentDocumentId": "0695h00000JAKrzAAH",
6   "Title": "solutionpdf"
7 }
```

Request 3: (POST): "Create ContentDocumentLink"

Postman interface showing a POST request to the Salesforce API endpoint: `https://meri-dev-ed.trailblaze.my.salesforce.com/services/data/v50.0/objects/ContentDocumentLink/`. The request is part of a collection named "SFDC Integration(External system)".

The request body is defined in the "Body" section, displaying a JSON object:

```
1 {
2   "ContentDocumentId": "0695h00000JAKrzAAH",
3   "LinkedEntityId": "0695h00000JevIAAE"
4 }
```

The response is shown in the "Body" section, displaying a JSON object:

```
1 {
2   "id": "06A5h00000QlcN6EAL",
3   "success": true,
4   "errors": []
5 }
```


Teddy Kimbler | Salesforce

meri-dev-ed.trailblaze.lightning.force.com/lightning/r/Contact/0035h00000lvEvIAAE/view

Search...

☆

+

🏠

?

⚙️

🔔

👤

Use Cases

Home

Contacts

Chatter

Cricket Data

Contact

Mr. Teddy Kimbler

+ Follow

New Contact

New Case

New Note

View All

Campaign History (0)

Add to Campaign

Notes & Attachments (1)

Upload Files

solutionpdf

Feb 3, 2023 • 556KB • pdf

View All

consumer (0)

New

Approval History (0)

Markers (2)

ActivityChatter

New Task

Log a Call

New Event

More

Create a task...

Add

Filters: All time • All activities • All types

⚡ Youtube Player

Step 2:

Use Node JS and upload file using a JS program.

Environment file for Credentials

```
SF_LOGIN_URL = https://login.salesforce.com
SF_USERNAME = usernameXXXX@usernameXXXX.com
SF_PASSWORD = passwordHereXXXXX
SF_SECURITY_TOKEN = XXXXXXXXXXXXXXXXXXXXX
```

I have saved it as .env file

Program to upload file

Change FILE_PATH & LinkedEntityId according to your need and then run the code

```
// upload file and link it to record using firstpublishlocationid

// required
const jsforce = require("jsforce");
require("dotenv").config();
const fs = require("fs");

// required for multi-part file upload
var FormData = require("form-data");
const mime = require("mime-types");
const axios = require("axios");

// required credentials and parameters
const { SF_LOGIN_URL, SF_USERNAME, SF_PASSWORD, SF_SECURITY_TOKEN } =
  process.env;

// about file & record to link file to
FILE_PATH = "/Users/Dell/Desktop/HorseRide.mp4"; // file to be uploaded
const LinkedEntityId = "0035h00000lvEvIAAE"; // record Id of record. to which we want to link file

// main method that log into SFDC, uploads file and links it to a record
const main = async (
  SF_LOGIN_URL,
  SF_USERNAME,
  SF_PASSWORD,
  SF_SECURITY_TOKEN,
  FILE_PATH,
  LinkedEntityId
```

```

=> {
  const conn = new jsforce.Connection({
    loginUrl: SF_LOGIN_URL,
  });

  // log into salesforce
  console.log("logging in to salesforce ....");

  await conn.login(SF_USERNAME, SF_PASSWORD + SF_SECURITY_TOKEN, (err, res) => {
    if (err) {
      console.log(err);
    } else {
      console.log("logged in!");
      console.log(res);

      // read file size scope start
      console.log("reading file size ....");

      fs.stat(FILE_PATH, (err, stats) => {
        if (err) {
          console.error(err);
        } else {
          // read file size scope start

          fileSizeInBytes = stats.size;
          console.log(`File size: ${fileSizeInBytes} bytes`);

          if (fileSizeInBytes > 36000000) {
            // create multi-part form data to send file

            console.log("file size greater than 36 MB");

            const createFormData = (file) => {
              // File
              //console.log("File received:", file);

              // linking it with a record
              console.log("creating the multi-part form data ....");

              const contentVersion = {
                FirstPublishLocationId: LinkedEntityId, //Id to which the content version needs to be linked
                Title: FILE_PATH.replace(/^.*[\\\/]/, ""), //get file name from file path
                PathOnClient: FILE_PATH.replace(/^.*[\\\/]/, ""), //get file name from file path
                Origin: "H",
              };

```

```

const form = new FormData();
form.setBoundary("boundary_string");
form.append("entity_content", JSON.stringify(contentVersion), {
  contentType: "application/json",
});
form.append("VersionData", file, {
  filename: FILE_PATH.replace(/^.*[\\\/]/, ""), //get file name from file path
  contentType: mime.lookup(FILE_PATH.replace(/^.*[\\\/]/, "")), //get file name from file path
});
return form;
};

// read file, to create multi-part form data
console.log("reading file, to create multi-part form data ....");

fs.readFile(FILE_PATH, { encoding: "base64" }, (err, data) => {
  if (err) {
    console.error(err);
    return;
  } else {
    const formData = createFormData(data);

    // upload file as multi-part form data
    console.log("uploading as multi-part form data ....");

    const URL =
      conn.instanceUrl +
      "/services/data/v51.0/subjects/ContentVersion";

    axios({
      method: "post",
      maxContentLength: Infinity,
      maxBodyLength: Infinity,
      url: URL,
      headers: {
        Authorization: "Bearer " + conn.accessToken,
        "Content-Type": `multipart/form-data; boundary=\`"boundary_string"\``,
      },
      data: formData,
    });

    console.log("File Uploading ....");

    // uploading as multi-part scope over, inside if
  }
}

```

```

});
} else {
  // file size < 36 MB. hence, no need of multi-part form data
  // read file, convert to base64 scope start
  console.log("reading file, converting to base64 ....");

  fs.readFile(FILE_PATH, { encoding: "base64" }, (err, data) => {
    if (err) {
      console.error(err);
      return;
    } else {
      base64Data = data;
      //console.log("base64 Data:", base64Data);

      // upload file scope start
      console.log("uploading file ....");

      conn.subject("ContentVersion").insert(
        {
          Title: FILE_PATH.replace(/^[^\\V]/, ""), //get file name from file path
          PathOnClient: FILE_PATH.replace(/^[^\\V]/, ""), //get file name from file path
          VersionData: base64Data,
          FirstPublishLocationId: LinkedEntityId, //Id to which the content version needs to be linked
          Origin: "H",
        },
        (err, res) => {
          if (err) {
            console.log("error:", err);
          } else {
            // file uploaded
            console.log("File uploaded!");
            console.log(res);

            // file uploading scope over, inside else
          }
        }
      );

      // reading file and converting to base64 scope over, inside else
    }
  });
} // else scope i.e if file size < 36 MB. that scope over

// read file size scope over
}

```

```

});

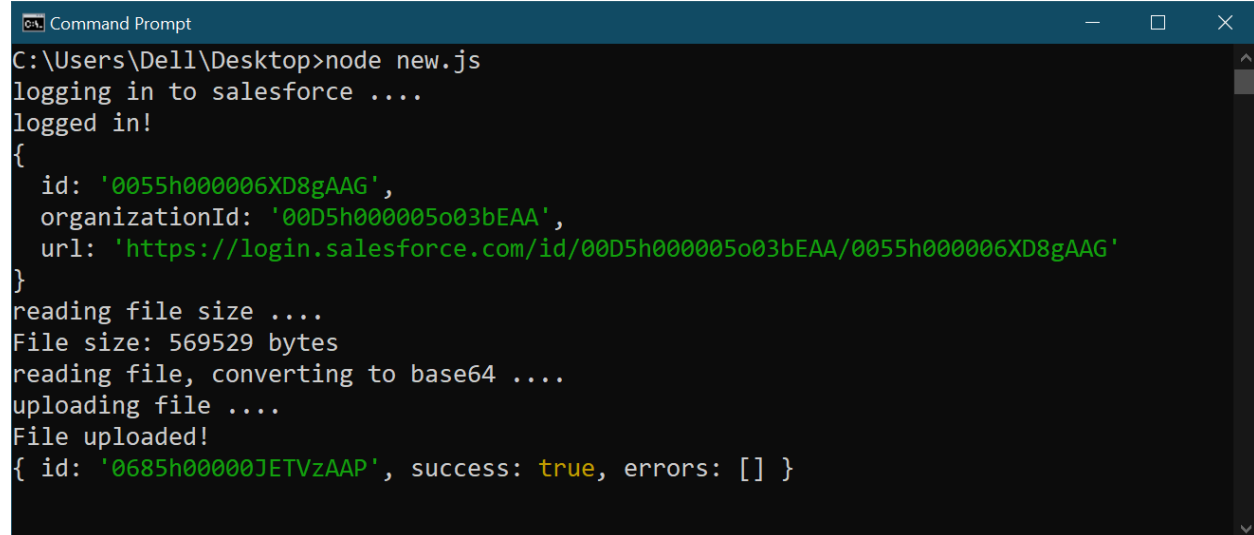
    // logged in scope over
}
});
}; // main method scope over

// calling main method with all the required parameters
main(
    SF_LOGIN_URL,
    SF_USERNAME,
    SF_PASSWORD,
    SF_SECURITY_TOKEN,
    FILE_PATH,
    LinkedEntityId
);

```

I have installed NodeJS and I have saved the code as “new.js”

Executing code for payload less than 36 MB



```

C:\Users\Dell\Desktop>node new.js
logging in to salesforce ....
logged in!
{
  id: '0055h000006XD8gAAG',
  organizationId: '00D5h000005o03bEAA',
  url: 'https://login.salesforce.com/id/00D5h000005o03bEAA/0055h000006XD8gAAG'
}
reading file size ....
File size: 569529 bytes
reading file, converting to base64 ....
uploading file ....
File uploaded!
{ id: '0685h00000JETVzAAP', success: true, errors: [] }

```

Executing code for payload size greater than 36 MB

```
Command Prompt
C:\Users\Dell\Desktop>node new.js
logging in to salesforce ....
logged in!
{
  id: '0055h000006XD8gAAG',
  organizationId: '00D5h000005o03bEAA',
  url: 'https://login.salesforce.com/id/00D5h000005o03bEAA/0055h000006XD8gAAG'
}
reading file size ....
File size: 45708474 bytes
file size greater than 36 MB
reading file, to create multi-part form data ....
creating the multi-part form data ....
uploading as multi-part form data ....
File Uploading ....
```

Notes:

- The default limit for Salesforce Classic is 25MB. Support can increase the limit up to 65MB. Attachments larger than 36MB can be attached only via User Interface.
- This limit increase does not apply to OnDemand Email-to-Case as that 25mb limit is hard coded
- The Salesforce Lightning attachment limit is set at 2GB (a hard-coded limit).
- This limit controls the maximum size of an attachment when uploaded using the standard related list, 'Notes & Attachments.'

References: