COLLEAGUE REFERENCE / PUBLIC API

DLP-TEST
API Upload Guide (Private)

How to send synthetic files and payloads to dlp-test.com, obtain upload receipts, and interpret the results.

START HERE Public endpoint: https://www.dlp-test.com/api_priv/v1/uploads | API key: optional | Maximum request body: 500,000 bytes (decimal 500 KB).

What this service is for

DLP-TEST provides an external HTTPS destination for validating how an organisation's Data Loss Prevention (DLP) controls treat application-to-application uploads. It can be called by scripts, HTTP clients, test tools, and enterprise applications.

The API confirms whether it received and stored an upload. It does not independently determine whether a DLP policy inspected or allowed the transfer; that evidence comes from your organisation's security tools.

Quick-start checklist

- Prepare a small file containing synthetic (dummy) test data only.
- Generate a new random UUID v4 and remove its hyphens (32 hexadecimal characters).
- POST one file or payload to /api_priv/v1/uploads, including X-Test-Id.
- For raw/JSON uploads, provide X-File-Name if you want the original filename retained.
- Record the response and query /api_priv/v1/receipts/{testId} within the configured retention window.

Where to go in this guide

Section	What you will find
1–2	Endpoint reference, request rules and headers
3–5	Working examples: PowerShell, curl, Postman and Python
6–7	JSON and text uploads; receipts and HTTP status codes
8–9	DLP test interpretation, limits, safety and troubleshooting

Version covered: DLP-TEST API v1 (flat-file storage). Prepared 20th November 2026 from the v1 source package; examples assume the production URL shown above.

1 API endpoints and request lifecycle

Method	Full URL	Purpose
GET	https://www.dlp-test.com/api_priv/health	Check service status and deployed API version
POST	https://www.dlp-test.com/api_priv/v1/uploads	Upload one file or raw payload
GET	https://www.dlp-test.com/api_priv/v1/receipts/{testId}	Retrieve an unexpired receipt for an upload

How a test works

1. Client sends HTTPS POST with X-Test-Id and a payload.
2. API checks optional credentials, request size, source-IP quotas and available storage.
3. API streams data to the configured protected directory and computes SHA-256.
4. API returns HTTP 200 with a receipt containing the stored filename, byte count and hash.
5. Client or analyst looks up the receipt and correlates it with DLP policy events.

2 Upload request rules

Item	Requirement / behaviour
Transport	HTTPS; POST /api_priv/v1/uploads.
Content	One non-empty file, raw bytes, JSON or plain text. Multipart supports exactly one file part, with no other parts.
Test ID	Required X-Test-Id: random UUID v4 formatted as 32 hexadecimal characters with no hyphens. Generate a fresh ID for every upload.
Filename	Raw/JSON: use X-File-Name or a Content-Disposition filename. Multipart: uses the uploaded file part's filename.
Authentication	Anonymous requests allowed. If issued an individual key, send X-API-Key or Authorization: Bearer <key>.
Size	Maximum entire HTTP request body 500,000 bytes. Multipart headers and boundaries count against the limit.
Timing	Uploads time out after approximately 30 seconds in application processing.
Retention	Successful upload files are retained until the operator's separate cleanup routine runs; API has no file download endpoint.

IMPORTANT Do not test with live confidential, personal, customer or credential data. Use dummy files that imitate the structure or markers your DLP policies are intended to detect.

2.1 HTTP header reference

Header	Required?	What it does
X-Test-Id	Yes, POST	Random UUID v4 as 32 hex digits without hyphens, e.g. 4b4d0ac3d6ea44c98c20a238032ee0a5.
X-File-Name	Raw / JSON: recommended	Original filename for raw body uploads. Not required for multipart, which carries its filename.
Content-Type	Yes in practice	application/octet-stream, application/json, text/csv, text/plain or multipart/form-data with boundary.
X-API-Key	Optional	Individually issued API key. If sent and incorrect, request returns 401. Never publish personal keys.
Authorization	Alternative	Bearer <key> instead of X-API-Key.
Content-Disposition	Optional	Alternative filename carrier for raw uploads; browser tools may use filename* for Unicode.

2.2 Filename behaviour

The API removes unsafe path elements and characters from supplied names. It stores files directly in its configured upload directory, without daily or per-upload subdirectories. It never overwrites an existing file.

```
sample.csv        → sample.csv
sample.csv again  → sample_1.csv
sample.csv again  → sample_2.csv
(no filename)     → <generated-file-id>.upload
```

The receipt field fileName contains the exact sanitised name actually saved. Each upload also receives a separate fileId. Filenames are limited to 160 characters and may be modified for Windows filesystem safety.

2.3 Common identifiers

testId is chosen by the client and is needed to look up the receipt. fileId is allocated by the server. A reused testId in the same receipt scope may result in HTTP 409 Conflict. Do not reuse an ID when retrying a new upload attempt.

```
PowerShell:  [guid]::NewGuid().ToString("N")
Python:      uuid.uuid4().hex
```

RECEIPT ACCESS If you upload using an issued API key, supply that same key when retrieving the receipt. Anonymous uploads and separately keyed uploads use different receipt scopes.

3 Windows PowerShell: upload a file

Recommended for Windows-based testing. Works with Windows PowerShell 5.1 and PowerShell 7. It sends the file as the raw HTTP body and preserves its filename using X-File-Name.

```
$path = 'C:\DLPTests\synthetic-sample.csv'
$url = 'https://www.dlp-test.com/api_priv/v1/uploads'

if (-not (Test-Path -LiteralPath $path -PathType Leaf)) {
    throw "File not found: $path"
}
if ((Get-Item -LiteralPath $path).Length -gt 500000) {
    throw 'File exceeds 500,000 bytes'
}

$testId = [guid]::NewGuid().ToString('N')
$headers = @{
    'X-Test-Id'   = $testId
    'X-File-Name' = [IO.Path]::GetFileName($path)
}
# Optional: $headers['X-API-Key'] = $env:DLPTEST_API_KEY

$response = Invoke-RestMethod -Uri $url -Method POST `
    -Headers $headers -InFile $path `
    -ContentType 'application/octet-stream' -ErrorAction Stop

$response | Format-List
Write-Host "Keep this Test ID: $testId"
```

Look up the matching receipt

```
$receiptUrl = "https://www.dlp-test.com/api_priv/v1/receipts/$testId"
$receipt = Invoke-RestMethod -Uri $receiptUrl `
    -Method GET -Headers $headers
$receipt | Format-List
```

For a fresh upload, run the block again to generate a new test ID. If the original filename already exists, the returned fileName will include _1, _2 and so on. The file may be inspected by the website operator, but cannot be downloaded through the API.

POWERSHELL GOTCHA -InFile does not automatically send the local filename. Without X-File-Name, expect a generated .upload filename. Avoid joining two pasted PowerShell commands onto the same line.

4 curl: raw and multipart uploads

On Windows PowerShell use curl.exe (not curl, which may be an alias). The examples below use PowerShell to generate a valid test ID.

4.1 Raw file upload

```
$id = [guid]::NewGuid().ToString('N')
curl.exe -i -X POST `
  'https://www.dlp-test.com/api_priv/v1/uploads' `
  -H "X-Test-Id: $id" `
  -H 'X-File-Name: synthetic-sample.csv' `
  -H 'Content-Type: text/csv' `
  --data-binary '@C:\DLPTests\synthetic-sample.csv'
```

4.2 Multipart/form-data upload

```
$id = [guid]::NewGuid().ToString('N')
curl.exe -i -X POST `
  'https://www.dlp-test.com/api_priv/v1/uploads' `
  -H "X-Test-Id: $id" `
  -F 'file=@C:\DLPTests\synthetic-sample.csv'
```

For multipart, curl sets the Content-Type and boundary; do not manually set multipart/form-data without the boundary. The file part transmits the filename automatically. Include exactly one file part, and keep the full multipart request below 500,000 bytes.

4.3 Receipt lookup

```
curl.exe -i "https://www.dlp-test.com/api_priv/v1/receipts/$id"
```

5 Postman: no-code API testing

- Create a new POST request to https://www.dlp-test.com/api_priv/v1/uploads.
- On the request's Pre-request Script tab, generate a fresh test ID using the script below.
- Add header X-Test-Id with value {{testId}}. Add X-API-Key only if a private key was issued.
- For a file, choose Body → form-data, name one field file, set its type to File and choose your synthetic file. Do not manually set the multipart Content-Type.
- Alternatively choose Body → binary and add X-File-Name, or Body → raw → JSON with Content-Type application/json and X-File-Name.
- Send the request, record the status and receipt. In a separate GET request, use /api_priv/v1/receipts/{{testId}} to query the result.

```
const id = pm.variables.replaceIn("{{$guid}}").replace(/-/g, "");
pm.collectionVariables.set("testId", id);
```

Attach the script to the POST request only; if it also runs on the GET receipt request, it will generate a different ID. Postman's dynamic {{$guid}} value should yield a version-4 UUID; check the generated value if your setup differs.

6 Python: automated HTTPS uploads

This example uses only the Python 3 standard library. It uploads a synthetic CSV, prints the response, and checks the receipt. No external Python packages are required.

```
import json
import os
from pathlib import Path
from uuid import uuid4
from urllib.request import Request, urlopen
from urllib.error import HTTPError, URLError

base = 'https://www.dlp-test.com/api_priv/v1'
path = Path(r'C:\DLPTests\synthetic-sample.csv')
data = path.read_bytes()
if not data or len(data) > 500000:
    raise ValueError('File must be 1 to 500,000 bytes')

id = uuid4().hex
headers = {'X-Test-Id': id,
           'X-File-Name': path.name,
           'Content-Type': 'text/csv'}
if os.getenv('DLPTEST_API_KEY'):
    headers['X-API-Key'] = os.environ['DLPTEST_API_KEY']

print('Test ID:', id)
try:
    req = Request(base + '/uploads', data=data,
                  headers=headers, method='POST')
    with urlopen(req, timeout=35) as r:
        print('Upload:', json.loads(r.read()))
except (HTTPError, URLError) as exc:
    print('Upload error:', exc)

try:
    req = Request(base + '/receipts/' + id,
                  headers=headers, method='GET')
    with urlopen(req, timeout=35) as r:
        print('Receipt:', json.loads(r.read()))
except (HTTPError, URLError) as exc:
    print('Receipt not available:', exc)
```

RETRY BEHAVIOUR Always use a new test ID for a new POST attempt. Do not blindly retry failed uploads: some DLP products terminate connections after some data has already arrived. Check the receipt and DLP console first.

7 JSON, plain text and other content

The upload endpoint accepts a raw HTTP request body. It does not require a particular file format or parse JSON as a business object. Content-Type describes the payload for the receipt and for systems under test.

7.1 Synthetic JSON using PowerShell

```
$id = [guid]::NewGuid().ToString('N')
$headers = @{
    'X-Test-Id' = $id
    'X-File-Name' = 'synthetic-payload.json'
}
$body = '{"classification":"SYNTHETIC-DLP-TEST-ONLY",' +
        '"recordId":"DEMO-0001","note":"dummy content"}'

Invoke-RestMethod `
    -Uri 'https://www.dlp-test.com/api_priv/v1/uploads' `
    -Method POST -Headers $headers `
    -ContentType 'application/json' -Body $body
```

7.2 Plain-text payload

```
$id = [guid]::NewGuid().ToString('N')
Invoke-RestMethod `
  -Uri 'https://www.dlp-test.com/api_priv/v1/uploads' `
  -Method POST `
  -Headers @{ 'X-Test-Id'=$id; 'X-File-Name'='dummy.txt' } `
  -ContentType 'text/plain' `
  -Body 'SYNTHETIC-DLP-TEST-ONLY | Test marker 0001'
```

7.3 Other file formats

For binary documents, archives or images, use a raw file-body request or a single-file multipart request. Preserve the correct file extension via X-File-Name for raw uploads. The server stores the bytes without attempting to open, execute or unpack the file.

A client may use Content-Type such as application/pdf, application/zip, application/vnd.openxmlformats-officedocument.wordprocessingml.document or application/octet-stream. Content-Type affects reporting, not whether the API will accept a properly formed upload.

BROWSER UPLOAD OPTION The supplied API Upload Methods page provides multipart, raw binary and JSON modes on the same website origin. If a separate website tries to call the API using browser JavaScript, cross-origin access may fail because the v1 server does not configure CORS; use the same-origin page or a non-browser client.

8 Successful responses and receipt lookup

8.1 Example HTTP 200 response

```
{
  "fileId": "23d47604c9cc4f62b87be7e60b5702b6",
  "testId": "dec7ddd6d9d84dea8f0769ae9181024b",
  "receivedUtc": "2026-11-18T11:37:17.1415926+00:00",
  "bytesReceived": 508,
  "contentType": "text/csv",
  "bodySha256": "651befab1bad450bbcac...",
  "status": "stored",
  "fileName": "synthetic-sample.csv"
}
```

Illustrative fields and shortened sample hash; each live response contains a complete SHA-256 hash. bytesReceived represents the extracted file body (or the entire raw payload), not multipart framing.

Response field	Meaning
fileId	Server-generated upload identifier
testId	Client-provided UUID v4 for receipt correlation
receivedUtc	UTC time at which the API recorded the upload
bytesReceived	Number of payload/file bytes received
contentType	Media type for the stored payload/file
bodySha256	SHA-256 digest of stored content bytes
status	stored when upload completed and receipt was written
fileName	Actual on-disk filename after sanitisation and any numeric suffix

8.2 Retrieve the receipt

```
GET https://www.dlp-test.com/api_priv/v1/receipts/{testId}

# Example with PowerShell:
Invoke-RestMethod "https://www.dlp-test.com/api_priv/v1/receipts/$testId"
```

Receipt lookup returns HTTP 200 when found and unexpired, or HTTP 404 when not found or expired. Metadata retention is configurable by the service operator and defaults to 60 minutes. Receipts are not file download URLs. Use the same individual API key, if any, that was used for the upload.

9 HTTP status codes and troubleshooting

HTTP	Typical meaning	What to do
200	Upload stored / receipt found	Record testId, timestamp, hash and any DLP incident.
400	Invalid or empty request; invalid/missing UUID; multipart problems	Generate new UUID v4 in N format. Supply exactly one file. Check boundary and body.
401	Supplied API key is invalid	Remove the key for anonymous testing or use your issued key.
403	HTTPS required	Use an https:// URL.
404	Receipt missing or expired	Check exact testId, same key scope and retention. Not proof of a DLP block.
404.13	IIS request filtering rejected oversized body	Reduce body size to <= 500,000 bytes including multipart overhead.
408	Application upload processing timed out	Try a smaller synthetic payload or investigate slow connection.
409	testId already used in that receipt scope	Generate a fresh UUID before a new upload.
413	Request too large	Stay below the 500,000-byte total HTTP body limit.
429	Source-IP minute/day quota, global quota or concurrency limit	Read Retry-After where present; wait rather than rapidly retry.
500/503	Service or state error	Check service status; contact DLP-TEST operator if persistent.
507	Storage unavailable or insufficient free space	Contact site operator; do not resubmit continuously.

Frequently encountered issues

A file is saved as <GUID>.upload A raw upload did not include X-File-Name or Content-Disposition filename. Supply X-File-Name.

The same name becomes sample_3.csv Expected: original name already exists. Suffix increments to avoid overwrites.

PowerShell reports Format-List$headers Paste commands on separate lines; run the complete example as a script.

HTTP 409 after re-running a request Test ID was likely reused; generate a new v4 UUID for each POST.

Upload blocked locally but no HTTP response Check your DLP logs and then query the receipt; absence of a response alone does not identify the blocker.

10 Usage limits and responsible testing

Control	Public API v1 setting / effect
Maximum request body	500,000 bytes (decimal 500 KB). Multipart includes boundaries/headers; target <490 KB for a simple multipart file.
Per-IP rate	10 admitted upload attempts per rolling 60 seconds.
Per-IP daily	100 admitted upload attempts per UTC calendar day.
Global daily	2,000 admitted upload attempts across the entire API per UTC day.
Concurrency	2 concurrent uploads per source-IP partition
Source IP identity	IPv4 address, or IPv1 /64 prefix; users behind one corporate proxy may share a quota.
Failed transfers	Quota attempt is recorded before body read. An interrupted or failed transfer can still consume an allowance.
Receipt retention	Operator-configurable; current sample default 60 minutes. Physical file retention is independent.
Storage and cleanup	Successful files remain in protected server storage until operator cleanup; not publicly retrievable.

Safe test design

- Use only authorised synthetic payloads; never send real customer, payroll, source code, password or regulated data.
- Avoid stress-testing production daily limits. Coordinate planned high-volume testing with the DLP-TEST operator.
- Do not share issued API keys, include them in screenshots, or place them in Git repositories.
- Record a unique testId for each attempt, and preserve the UTC request time and the DLP alert/incident reference.
- Consider corporate NAT/proxy effects; a whole team may share the same public-IP quota.
- Do not assume an HTTP error means a DLP block; network, request-size, rate-limit and server failures can look similar.

DLP result interpretation

Client result	Receipt lookup	Suggested interpretation
HTTP 200	HTTP 200	Destination received payload. Check DLP policy log for allow/monitor actions.
Error or connection drop	HTTP 200	Destination may still have received full upload; investigate DLP and client/network behaviour.
Error or connection drop	HTTP 404 / unavailable	Inconclusive; may be blocked, rejected, expired or unreachable. Correlate logs.
HTTP 429 / 413	No receipt	API or IIS likely rejected request by quota or size; not a DLP-policy verdict.

11 Suggested test record and support details

For repeatable DLP validation, capture a short record for each upload attempt. Do not include genuine sensitive source data in a ticket or shared spreadsheet.

Field	Example / instruction
Purpose	Confirm outbound API upload inspection for a synthetic CSV
Date/time (UTC)	Record local UTC time of attempt and any receipt time
Source / network	Corporate endpoint, proxy, network egress or workload identity
Method	PowerShell raw POST / curl multipart / Postman JSON
Endpoint	https://www.dlp-test.com/api_priv/v1/uploads
testId	32-character UUID v4; new for each attempt
Filename / byte count	Synthetic filename; size <=500,000-byte total request
Client HTTP result	200, 429, timeout, blocked, connection reset, etc.
Receipt fields	fileName, receivedUtc, bodySha256, status
DLP policy result	Blocked / monitored / allowed; include incident or event reference
Analyst conclusion	State evidence, uncertainty and any follow-up test

Quick reference

```
UPLOAD
POST  https://www.dlp-test.com/api_priv/v1/uploads
Headers: X-Test-Id: <new UUID v4, no hyphens>
         X-File-Name: <filename>  (raw/JSON only)
         X-API-Key: <issued key>   (optional)
Body:    raw bytes, JSON, text or ONE multipart file

RECEIPT
GET   https://www.dlp-test.com/api_priv/v1/receipts/<testId>
Header: X-API-Key: <same issued key, if upload used one>

HEALTH
GET   https://www.dlp-test.com/api_priv/health
```