

DROP Technical and API Reference Documentation

The information below will assist you with building your integration, whether API-based or manual, with DROP for purposes of deletion request processing. These documents are provided for guidance only. You are responsible for ensuring your own compliance and accuracy.

Please be aware that this document is not exhaustive and that your business is responsible for ensuring compliance with all applicable statutory and regulatory requirements. Read the full text of the regulations at coppa.ca.gov/regulations.

Download API Specification (OpenAPI YAML)

Data brokers are required to access DROP to process deletion requests beginning on August 1, 2026. The documentation below provides technical requirements and information to facilitate testing in the sandbox and integration with production. Confirm you are using the latest version of the documentation. **Contact us** if you have any questions.

Contents

1. Getting Started
2. Integration Workflow
3. Working With the Data
4. API Operations
5. Reference

Getting Started

You may build your integration using the DROP Data Broker API or by using the manual download/upload option. For either option, you need to complete the steps below.

1. **Download** hashed consumer deletion list(s) from DROP.
2. **Match** the hashed identifiers in those list(s) against your own records.
3. **Upload** a status file indicating whether each record was deleted, opted out, exempted, or not found.

This guide provides information about the integration workflow, working with the data in DROP, and more. If you elect to build your integration via API, you can use the information in this document to confirm account setup, authentication, and processing via API.

Who this documentation is for

- Data brokers
- Engineering teams building automated download-and-upload integrations
- Quality assurance, operations, and compliance staff supporting the integration lifecycle

Prerequisites for integration

CalPrivacy DROP Technical and API Reference Documentation

Before you begin, ensure you have an account in DROP and have registered (if applicable) and paid any required fees.

If you elect to integrate via API, please also ensure you have the following:

Requirement	Details
API key	Generated after selecting your consumer deletion list(s)
HTTPS client	Ability to make <code>GET</code> and <code>POST</code> requests over TLS 1.2+ (for example, curl, PowerShell, or any HTTP library)
CSV handling	Ability to read and write CSV files following the required naming convention
Secret storage	Secure storage for API keys
Logging	Ability to log request timestamps and upload/download outcomes

API base URL

Environment	URL
Production	<code>https://api.drop.privacy.ca.gov</code>
Sandbox	<code>https://api.drop.privacy.ca.gov/sandbox</code>

Authentication

All API requests require an API key passed in the `X-API-KEY` header:

```
X-API-KEY: your-api-key-here
```

Your API key is scoped to the consumer deletion list(s) you select. Only the selected list(s) will appear in your downloads.

API key security

- Store keys securely using environment variables or secret management systems
- Regenerate keys if compromised or if list selection changes

Obtain an API key

1. Sign in to the [Data Broker Portal](#).
2. For production integration, navigate to **Home** → **Consumer Deletion Lists**.
3. Select the list(s) your organization will process.
4. Click **Save**.
5. Open the **API Key** tab.
6. Click **Get a new API key**.

Your key is now ready to use.

NOTE: During sandbox testing, navigate to **SANDBOX ENVIRONMENT** → **ISSUE SANDBOX API KEY**.

API quick start

1. Obtain an API key and confirm your selected consumer deletion list(s).
2. Verify connectivity with an authenticated request.
3. Download the current ZIP archive.
4. Extract and process the CSV files.

CalPrivacy DROP Technical and API Reference Documentation

5. Upload response CSV(s) using the original file name.
6. Inspect the JSON response for accepted and rejected files.

Step 1 — Verify connectivity

Send a request to confirm your API key works:

```
curl -X GET "https://api.drop.privacy.ca.gov/data/download" \  
  -H "accept: */*" \  
  -H "X-API-KEY: your-api-key-here" \  
  -I
```

A `200 OK` response confirms authentication.

- `401 Unauthorized` — incorrect API key. Correct the key or generate a new one and try again.
- `403 Forbidden` — registration and/or payment of required fees is incomplete. Confirm account status in the data broker portal and try again.

Step 2 — Download consumer data

```
curl -X GET "https://api.drop.privacy.ca.gov/data/download" \  
  -H "accept: */*" \  
  -H "X-API-KEY: your-api-key-here" \  
  --output download.zip
```

PowerShell alternative:

```
$apiKey = "your-api-key-here"  
$url = "https://api.drop.privacy.ca.gov/data/download"  
$headers = @{  
    "X-API-KEY" = $apiKey  
    "Accept"    = "text/zip"  
}  
Invoke-WebRequest -Uri $url -Headers $headers -OutFile "download.zip"
```

The response is a ZIP archive containing one CSV file per selected consumer deletion list.

Step 3 — Process records

Extract the ZIP and read each CSV. Match the hashed identifiers against your records and determine the appropriate action. See [Integration Workflow](#) for the full processing lifecycle.

Step 4 — Upload your response

Respond by uploading one or more CSV files (not the ZIP archive) containing `Id` and `Status` columns. Each response CSV must use the same file name as the corresponding file from your download.

If you need to upload multiple response files for the same downloaded list, add a suffix to the file name using an underscore followed by up to 10 alphanumeric characters (for example: `20260312_4821_Email_01.csv`). Uploading the same file name again without a suffix will result in an error.

You may upload a full or partial response in each file. If you upload only a partial response, you can provide the remaining records later by uploading additional files with the same base file name and different suffixes.

```
curl -X POST "https://api.drop.privacy.ca.gov/data/upload" \
-H "accept: */*" \
-H "X-API-KEY: your-api-key-here" \
-H "Content-Type: multipart/form-data" \
-F "files=@20260312_4821_Email_retry01.csv;type=text/csv" \
-F "files=@20260312_4821_Phone.csv;type=text/csv" \
-F "files=@20260312_4821_NDZ.csv;type=text/csv"
```

A successful response:

```
{
  "mode": "new",
  "acceptedCount": 3,
  "rejectedCount": 0,
  "accepted": [
    { "fileName": "20260312_4821_Email_retry01.csv", "message": "Accepted. NEW file queued for processing." },
    { "fileName": "20260312_4821_Phone.csv", "message": "Accepted. NEW file queued for processing." },
    { "fileName": "20260312_4821_NDZ.csv", "message": "Accepted. NEW file queued for processing." }
  ],
  "rejected": []
}
```

If you need to correct a previous submission, use the `/data/amend` endpoint. See [API Operations](#) for details.

Step 5 — Confirm successful integration

A successful integration via API indicates you are able to:

- Authenticate against the API
- Download and extract a ZIP archive
- Read CSV files and understand the schema
- Produce a correctly named response CSV with `Id, Status` columns
- Upload the response and receive confirmation

Confirm that you can complete all of the steps above prior to automating the end-to-end processing workflow.

Integration Workflow

Deletion request processing is an ongoing pull-process-respond cycle, not a one-time integration. The steps below summarize that cycle.

For more information on the legal requirements related to deletion request processing, including relevant timelines for ongoing processing, see California Civil Code section 1798.99.80 et seq. and California Code of Regulations (CCR), title 11, section 7600 et seq.

1. Set up account and register, if applicable (one-time)

2. Generate API key (one-time or as needed)

3. Download ZIP archive

4. Extract and read CSV(s)

5. Match against your records

6. Take required action for each match

7. Upload status CSV(s)

Repeat

Step by step

1. Set up your account (one-time)

Create an account in DROP, complete registration if applicable, pay required fees, and select the consumer deletion list(s) you will process.

2. Generate your API key (if integrating via API)

Generate an API key in the portal. The key is scoped to your selected consumer deletion list(s).

NOTE: Integration via API is not required. You may use the manual download/upload option instead.

3. Download data

Call `GET /data/download` to retrieve a ZIP archive containing one CSV file per selected consumer deletion list. Each CSV contains hashed consumer identifiers included in deletion requests.

4. Extract and read

Unzip the archive and read each CSV file. Preserve the exact file name — you will need it when uploading your response.

5. Match records

Compare each hashed identifier against your own records. The hashing and standardization rules are documented in [Hashing and Standardization](#).

6. Process records based on match outcomes

Use the following status codes when reporting match outcomes:

Code	Label	Meaning
2	Exempted	Match found and personal information is exempt
3	Deleted	Match found and non-exempt personal information was deleted
4	Opted out	Multiple consumers are linked to the same identifier and all were opted out of sale or sharing
5	Not found	No match found after completing the matching process

For more information on the legal requirements based on match outcomes, see California Civil Code section 1798.99.80 et seq. and CCR, title 11, section 7600 et seq.

NOTE: You must direct all service providers or contractors to process deletions and opt outs based on the match outcomes above.

7. Upload status response

Create a CSV with `Id,Status` columns, using the exact file name from the download. Upload it using:

- `POST /data/upload` for a first-time submission for a given run
- `POST /data/amend` for corrections to a previous submission

Validation

DROP validates the following for every upload:

- API key must be present and valid
- Uploads must use `multipart/form-data` with the field name `files`
- File name must match the downloaded file exactly
- CSV header must be `Id,Status`
- Status values must be `2`, `3`, `4`, or `5`

If validation fails, the API returns an HTTP error or a file-level rejection message.

NOTE: We will send an email to the primary and secondary contact email addresses listed on your account if there are errors with your upload. Carefully review the error details in the email, correct them, and resubmit.

Schedules and retries

Daily maintenance window

DROP closes push and pull requests between `1:00 AM` and `3:00 AM` Pacific Time daily to batch the most current data.

- **During this window:** Requests may fail. Do not treat these as permanent errors.
- **Outside this window:** Normal download and upload operations are available.

Ensure any automation schedules account for the daily maintenance window.

Confirmation

After uploading, check the response body:

```
{
  "mode": "new",
  "acceptedCount": 1,
  "rejectedCount": 0,
  "accepted": [
    { "fileName": "20260312_4821_Email.csv", "message": "Accepted. NEW file queued for processing." }
  ],
  "rejected": []
}
```

- `acceptedCount > 0` — your file is queued for processing.
- `rejectedCount > 0` — review the `rejected` array for error messages and correct the file before resubmitting.

Retry strategy

CalPrivacy DROP Technical and API Reference Documentation

Scenario	Action
Request during the daily maintenance window	Wait until after 3:00 AM PT, then retry
429 Too Many Requests	Wait 30 seconds and try again
500 Server Error	Retry later
401 Unauthorized	Fix or regenerate the API key and try again
403 Forbidden	Complete registration if applicable and pay all fees, then try again
Invalid file name or CSV header	Fix the file and resubmit. Resubmitting the same invalid file will fail again.

API failure

If you are unable to use the API for any reason, you must use the manual upload and/or download option in the portal.

Working With the Data

Understanding the data

Each downloaded file contains consumer deletion request data. Every row includes an `ID` column that identifies a single work item. The corresponding `Hash` or `ConcatenatedHash` column contains the hashed consumer identifier you will compare against your own records.

Your response file should include that same work item identifier in the `Id` column alongside the status you assign.

Consumer deletion lists

DROP delivers consumer deletion request data organized by list type. There are six list types, and each list includes one or more consumer identifiers. When you generate your API key, you select which list(s) you will process. The download will contain one CSV file per selected list.

Data brokers must select every consumer deletion list containing a consumer identifier corresponding to personal information about that consumer within the data broker's own records. However, a data broker may select fewer lists if the consumer identifiers used across multiple lists would lead to matches with the exact same set of consumers in the data broker's records.

Code	Identifier	Type
<code>NDZ</code>	First Name + Last Name + Date of Birth + ZIP	Concatenated hash
<code>Email</code>	Email address	Single-field hash
<code>Phone</code>	Phone number	Single-field hash
<code>MAID</code>	Mobile advertising ID	Single-field hash
<code>NameVIN</code>	First Name + Last Name + VIN	Concatenated hash
<code>CTVID</code>	Connected TV identifier	Single-field hash

Consumer data properties

Property	Format	Details
Name	Separate first name and last name fields	Names are collected, standardized, and hashed separately
Date of Birth	<code>YYYYMMDD</code>	Eight digits, four-digit year
ZIP Code	Up to 5 alphanumeric characters	First five characters after stripping non-alphanumeric characters
Email	String	Validated as a valid email address
Phone	Last 10 digits only	Strip non-numeric characters, then keep the last 10 digits
MAID	32 hexadecimal characters	Mobile Advertising ID
VIN	17 alphanumeric characters	Vehicle Identification Number
CTVID	8 to 32 alphanumeric characters	Unique device identifier for smart TVs, streaming players, and gaming consoles

File naming convention

Downloaded and uploaded CSV files share the same naming convention. You must use the exact file name from the download when uploading your response. Optionally, you may append an underscore followed by up to 10 alphanumeric characters to differentiate multiple uploads.

```
<YYYYMMDD>_<DataBrokerId>_<DataType>[_<OptionalSuffix>].csv
```

CalPrivacy DROP Technical and API Reference Documentation

Field	Description
<code>YYYYMMDD</code>	File date
<code>DataBrokerId</code>	Your short broker identifier, currently 4 digits. This ID is provided in the file name you download
<code>DataType</code>	List type code: <code>NDZ</code> , <code>Email</code> , <code>Phone</code> , <code>MAID</code> , <code>NameVIN</code> , or <code>CTVID</code>
<code>OptionalSuffix</code>	Optional. Up to 10 alphanumeric characters to differentiate uploads

Example file names:

```
20260312_4821_NDZ.csv
20260312_4821_Email.csv
20260312_4821_NameVIN.csv
20260312_4821_Email_part01.csv
```

File encoding and formats

- CSV encoding: UTF-8
- Hash input encoding: UTF-8
- Hash output format: Base64

The download response is a ZIP archive containing CSV files. Uploads are individual CSV files sent via `multipart/form-data`.

Downloaded file schema

The CSV schema depends on the list type.

Single-field lists (Email, Phone, MAID, CTVID)

Column	Description
<code>ID</code>	Work item identifier
<code>Hash</code>	SHA-256 Base64 hash of the consumer identifier

Example (Email):

```
ID,Hash
679,KA18MT/ph6IHYjzT9zwETySDQyvSh87YuoSBpOQtKhE=
```

Concatenated-hash lists (NDZ, NameVIN)

Column	Description
<code>ID</code>	Work item identifier
<code>ConcatenatedHash</code>	SHA-256 Base64 hash of the concatenated fields

Example (NDZ):

```
ID,ConcatenatedHash
679,PQOfn1RffEKmqMmNAzDKKaoZCwxWbQZkQzPWmQo9REA=
```

Upload file schema

CalPrivacy DROP Technical and API Reference Documentation

Your response file must use the header `Id,Status` and the exact file name from the download.

Column	Description
<code>Id</code>	Work item identifier from the downloaded file
<code>Status</code>	Numeric status code

Example:

```
Id,Status
679,2
680,5
681,4
```

Status codes

Code	Label	Meaning
2	Exempted	Match found and all personal information is exempt
3	Deleted	Match found and non-exempt personal information was deleted
4	Opted out	Multiple consumers are linked to the same identifier and all were opted out of sale or sharing
5	Not found	No match found after completing the matching process

For more information on the legal requirements based on match outcomes, see California Civil Code section 1798.99.80 et seq. and CCR, title 11, section 7600 et seq.

Hashing and standardization

All consumer identifiers in DROP are SHA-256 hashed. Before hashing, you must standardize each value according to the rules below.

General rules

1. Apply the identifier-specific standardization described below.
2. Hash with SHA-256 using UTF-8 input encoding.
3. Output as Base64.

Tip: A SHA-256 Base64 hash is typically 44 characters and commonly ends with `=`.

Name standardization

- Replace accented Latin characters with plain ASCII equivalents
- Transliterate Greek and Cyrillic characters to Latin equivalents
- Leave Chinese, Japanese, Korean, Arabic, and Hebrew characters unchanged
- Treat compound first names as a single unit (e.g., `Juan Pablo` → `juanpablo`)
- Remove hyphens, apostrophes, and spaces
- Use lowercase values

Input	Standardized	SHA-256 Base64
Juan Pablo	<code>juanpablo</code>	<code>91hIbrbzNeqHs3o81O5yNrXUj7wDd2shvZ6THKi9qz8=</code>
Martinez	<code>martinez</code>	<code>2wRPGbwBNxhShjRczx8GfS2c4cjvs4NJskeWloUNtp8=</code>

Date of birth standardization

CalPrivacy DROP Technical and API Reference Documentation

Convert to `YYYYMMDD` format. Use a four-digit year.

Input	Standardized	SHA-256 Base64
July 4, 1776	17760704	skXYXxBER6HQTZ3rXSZH1wVGLQ054mS5rbR/bwvzy4I=

ZIP code standardization

Keep alphanumeric only, use only lowercase values, remove leading zeros, take first 5 characters.

Input	Standardized	SHA-256 Base64
91790-3771	91790	2FPZucR4x7U8KlM+SFAX4LPghwNz/PIZUCSudDh0o/s=
M1B 1A1	mlb1a	n8L9q8mVeT6Xt9/EeUNiTukGDrkbPJ3DvOEx14uElxk=
712345	71234	aeNUYKh7Xw5sqpxSbSP9eOHsj6iXewbUyavv89DIuhQ=

Email standardization

Only trim whitespace and convert to lowercase. Do not remove dots, plus signs, or other characters.

Input	Standardized	SHA-256 Base64
Anna.Smith@Domain.com	anna.smith@domain.com	KA18MT/ph6IHYjzT9zweTySDQyvSh87YuoSBpOQtKhE=
danielle.johnson12@example.com	danielle.johnson12@example.com	mKDnDvwF2inxrKcK1hJN2TRkxPfl6kzNNTtU12eH8Bw=

Phone standardization

Strip all non-numeric characters. Use the last 10 digits, or all digits if fewer than 10 are present.

Input	Standardized	SHA-256 Base64
+1 (415) 555-9317	4155559317	vGM7y5n+hBXRSEAk1hHDPCbysyNgYTmXdMcagGUOY8E=
+84 (90) 123 4567	4901234567	ptzVkgbv9DonwvPCHmXmJ2SEoaolSh37z3ZzY/Gmm+U=
+354 (123) 4567	3541234567	Btrzydf5K6ALAKKXJGFHSx7u5bDzHC9WlVYtpq1n2rY=
5551273811	5551273811	jr/RAWYVN+ODBF2vRxbASPWio4x27OGI1y3IDhcwLo=

VIN standardization

Keep only alphanumeric characters and convert to lowercase.

Input	Standardized	SHA-256 Base64
1HGCM82633A004352	1hgcm82633a004352	iNswylm+OVSt8jAfFrvaiQ1R/0HAbgSwNGkwqo6QBss=

MAID standardization

Keep only hexadecimal characters (`0-9`, `a-f`) and convert to lowercase.

Input	Standardized	SHA-256 Base64
a3f1c2d4-5678-90ab-cdef-1234567890ab	a3f1c2d4567890abcdef1234567890ab	250KY6lOgzYUB3EhrkbDCE2kMEZQE69SF38muhoDudI=

CTVID standardization

Keep only alphanumeric characters and convert to lowercase. Minimum length after standardization: 8. Maximum length: 32.

Input	Standardized	SHA-256 Base64
B7e4f9a1-2345-6789-ABCD-ef0123456789	b7e4f9a123456789abcdef0123456789	MyXj2jIgpIH1LGsXl/fzPSswbsIl2rKPct7GOYbnIY=

Composite hashing

The `NDZ` and `NameVIN` list types use composite hashing — a multi-step process that hashes individual fields first, then hashes the concatenated result.

NDZ: Name + Date of Birth + ZIP

1. Standardize and hash each field separately (SHA-256, UTF-8, Base64).
2. Concatenate the four Base64 hashes in this order: `FirstName + LastName + DOB + ZIP`.
3. Hash the concatenated string using the same method.

Example: Danielle Johnson, DOB July 4, 1985, ZIP 91790.

Field	Standardized	SHA-256 Base64
First Name	danielle	5dUD1FgiKcTJq+JQ5JZUdlyIXrSbtJ338YYbt5/HNG4=
Last Name	johnson	K+TjOqPiH2/3rRRpj9WCKKHM47UDQLSAX/DGNIDuxIg=
DOB	19850704	IWi7qxOAbBJe0fNciDj76Eg84gmj40rB7aNMK/VnFOI=
ZIP	91790	2FPZucR4x7U8K1M+SFAX4LPghwNz/PIZUCSudDh0o/s=

Final hash: `PQOfn1RffEKmqMmNAzDKKaoZCwxWbQZkQzPWmQo9REA=`

NameVIN: Name + VIN

1. Standardize and hash each field separately (SHA-256, UTF-8, Base64).
2. Concatenate the three Base64 hashes in this order: `FirstName + LastName + VIN`.
3. Hash the concatenated string using the same method.

For consumers with multiple VINs, generate all Name × VIN combinations.

Example: Eve Genesis, VIN `1HGCM82633A004352`.

Field	Standardized	SHA-256 Base64
First Name	eve	hSYq33RRi7twx8uUzWfZ2RZp5age3x7+vVQ+rb2p+is=
Last Name	genesis	ruutSnlvzC4V3ExgYbRe2bNz8mrfx5jKfS2MxYGCCY4=
VIN	1hgcm82633a004352	iNswylm+0Vst8jAfFrvaIQ1R/0HAbgSwNGkwqo6QBss=

Final hash: `rtnDuXIe63jXYQQXW5r07GJ7lSsrib8+46QuKFwkOmk=`

API Operations

The canonical machine-readable reference is [openapi.yaml](#). Load it in Postman, Swagger, or any OpenAPI-compatible tool for an interactive view.

Endpoints

Method	Path	Description
GET	/data/download	Download consumer deletion request data as a ZIP archive
POST	/data/upload	Upload a new status response file
POST	/data/amend	Amend a previously uploaded response file

Base URL: `https://api.drop.privacy.ca.gov`

Download

GET /data/download

Retrieves a ZIP archive containing one CSV file per selected consumer deletion list.

Request headers:

```
X-API-KEY: your-api-key-here
Accept: text/zip
```

curl example:

```
curl -X GET "https://api.drop.privacy.ca.gov/data/download" \
  -H "accept: */*" \
  -H "X-API-KEY: your-api-key-here" \
  --output download.zip
```

PowerShell example:

```
$apiKey = "your-api-key-here"
$url = "https://api.drop.privacy.ca.gov/data/download"
$headers = @{
    "X-API-KEY" = $apiKey
    "Accept"    = "text/zip"
}
Invoke-WebRequest -Uri $url -Headers $headers -OutFile "download.zip"
```

Response: binary ZIP file containing CSV files such as:

```
20260312_4821_NDZ.csv
20260312_4821_Email.csv
```

Upload

There are two upload endpoints. Both accept the same `multipart/form-data` request. Only the mode differs.

POST /data/upload

CalPrivacy DROP Technical and API Reference Documentation

Upload a response file for the first time for a given downloaded file.

POST /data/amend

Upload a correction or update to a previously submitted response.

Request format

Both endpoints use `multipart/form-data` with the field name `files`.

Request headers:

```
X-API-KEY: your-api-key-here
Content-Type: multipart/form-data
Accept: */*
```

curl example (new):

```
curl -X POST "https://api.drop.privacy.ca.gov/data/upload" \
-H "accept: */*" \
-H "X-API-KEY: your-api-key-here" \
-H "Content-Type: multipart/form-data" \
-F "files=@20260312_4821_Email.csv;type=text/csv"
```

curl example (amend):

```
curl -X POST "https://api.drop.privacy.ca.gov/data/amend" \
-H "accept: */*" \
-H "X-API-KEY: your-api-key-here" \
-H "Content-Type: multipart/form-data" \
-F "files=@20260312_4821_Email.csv;type=text/csv"
```

Response format

```
{
  "mode": "new",
  "acceptedCount": 1,
  "rejectedCount": 0,
  "accepted": [
    { "fileName": "20260312_4821_Email.csv", "message": "Accepted. NEW file queued for processing." }
  ],
  "rejected": []
}
```

Field	Type	Description
<code>mode</code>	string	<code>"new"</code> or <code>"amend"</code>
<code>acceptedCount</code>	integer	Number of files accepted
<code>rejectedCount</code>	integer	Number of files rejected
<code>accepted</code>	array	Files successfully queued, each with <code>fileName</code> and <code>message</code>
<code>rejected</code>	array	Files rejected, each with <code>fileName</code> and <code>message</code> describing the error

Rejection messages

CalPrivacy DROP Technical and API Reference Documentation

Message	Cause	Fix
Invalid CSV header. Expected: Id,Status	CSV header row does not match schema	Make the first row exactly Id,Status
You already uploaded a file with the same filename for this run.	A file with the same name was already submitted for this run	Use a unique optional suffix in the file name and upload again, for example _retry01

Error handling

HTTP response codes

Code	Meaning	Remediation steps
200	Success	API request submitted successfully
400	Bad request	Check file formatting, file name, and CSV columns. Fix before retrying
401	Unauthorized	Incorrect API key. Correct or regenerate it in the Data Broker Portal before trying again
403	Forbidden	Complete registration if applicable and pay all fees, then try again
404	Not found	The endpoint URL cannot be found. Correct the path and try again
429	Too many requests	Wait 30 seconds before sending another API call
500	Server error	Server may be unavailable or in a maintenance window. Retry later

Retry guidance

Scenario	Retry?	Notes
429 Too Many Requests	Yes	Wait 30 seconds and try again
500 Server Error	Yes	Retry later
Request during maintenance window	Yes	Retry after 3:00 AM PT
401 Unauthorized	No	Regenerate or correct the API key first
403 Forbidden	No	Complete registration if applicable and pay all fees, then try again
404 Not Found	No	Fix the URL and try again
400 Bad Request	No	Fix the file or request and try again
Duplicate file name rejection	No	Use a unique optional suffix in the file name and upload again
Invalid CSV header rejection	No	Fix the header to Id,Status

Rate limits and schedules

- Push and pull requests are closed between 1:00 AM and 3:00 AM Pacific Time daily.
- Rate limits may be adjusted automatically based on system load.
- Schedule automated jobs outside the maintenance window and use exponential backoff for 429 Too Many Requests.

Reference

Glossary

Term	Definition
DROP	Delete Request and Opt-out Platform — the platform created by CalPrivacy for consumers to submit deletion and opt-out requests and for data brokers to retrieve and respond to them
Delete Act	California law requiring data brokers to process consumer deletion requests through DROP (see Civil Code section 1798.99.80 et seq.)
CPPA / CalPrivacy	The California Privacy Protection Agency — the state agency responsible for DROP
API key	The credential sent in the <code>X-API-KEY</code> header to authenticate API requests
Data broker ID	The unique identifier assigned to your organization, used in file naming
Consumer deletion list	A list containing one or more hashed consumer identifiers that consumers provided as part of a deletion request
Work item	A single consumer deletion request record identified by its <code>ID</code> in a CSV file
NDZ	Name + Date of Birth + ZIP — a concatenated-hash list type
MAID	Mobile Advertising Identifier
VIN	Vehicle Identification Number
CTVID	Connected TV Identifier — covers smart TVs, streaming players, and gaming consoles
NameVIN	Name + VIN — a concatenated-hash list type
New upload	The first submission of a response file for a given download run
Amend upload	A correction or update to a previously submitted response file
Composite hash	A multi-step hash where individual fields are hashed first, concatenated, then hashed again
Maintenance window	Daily <code>1:00 AM</code> to <code>3:00 AM</code> Pacific Time period when push and pull requests are closed

Status code reference

Code	Label	Meaning
2	Exempted	Match found and all personal information is exempt
3	Deleted	Match found and non-exempt personal information was deleted
4	Opted out	Multiple consumers are linked to the same identifier and all were opted out of sale or sharing
5	Not found	No match found after completing the matching process

HTTP response codes

CalPrivacy DROP Technical and API Reference Documentation

Code	Meaning	Remediation steps
200	Success	API request submitted successfully
400	Bad request	Check file formatting, file name, and CSV columns. Fix before retrying
401	Unauthorized	Incorrect API key. Correct or regenerate it in the Data Broker Portal before trying again
403	Forbidden	Complete registration if applicable and pay all fees, then try again
404	Not found	The endpoint URL cannot be found. Correct the path and try again
429	Too many requests	Wait 30 seconds and try again
500	Server error	Server may be down for maintenance. Retry later

File naming pattern

```
<YYYYMMDD>_<DataBrokerId>_<DataType>[_<OptionalSuffix>].csv
```

Documentation map

Document	Audience	Contents
Getting Started	All	Account setup, authentication, API key generation, quick start
Integration Workflow	All	The download-process-upload cycle, validation, retries
Working With the Data	Technical	List types, file schemas, standardization rules, hashing examples
API Operations	Technical	Endpoints, request examples, error handling, rate limits
Reference	All	Glossary, status codes, documentation map, and support information
OpenAPI Specification (YAML)	Developers	Machine-readable API definition for OpenAPI-compatible tools

Contact and support

Use the [Data Broker Portal](#) for account setup, credentials, and to [contact us](#).

When reporting an issue, include:

- Timestamp (Pacific Time)
- Endpoint and HTTP status code
- File name involved
- Upload response payload or error message
- Whether the request occurred during the 1:00 AM to 3:00 AM PT maintenance window

Version history

Version	Date	Notes
1.0.0	March 2026	Initial release