



Washington State
Liquor and Cannabis Board

Washington State Liquor and Cannabis Board

Cannabis Central Reporting System (CCRS)

Integrator API Guide

Connecting to CCRS and submitting files via the reporting API

Important Notice

This guide describes what is required to connect to the CCRS reporting API and submit files. It includes short examples of the standard requests involved, for reference.

It is the integrator's responsibility to design, build, test, secure, and maintain the software used to connect to CCRS. Any examples in this guide are for reference only. They show the standard OAuth 2.0 and HTTP requests involved, not a supported implementation.

The Washington State Liquor and Cannabis Board does not provide software development guidance, code support, integration support, or debugging for integrator systems, and does not assume responsibility for how an integrator builds or operates its connection to CCRS. Integrators are responsible for ensuring their systems function correctly and comply with all applicable requirements.

How it Works

CCRS accepts file submissions through a reporting API secured with machine-to-machine (M2M) authentication. Your software does two things, both using standard protocols:

1. **Obtain an access token** - send your issued credentials to the identity platform and receive a short-lived bearer token (standard OAuth 2.0 client-credentials flow).
2. **Submit a file** - send your data file to the CCRS API over HTTPS, including the bearer token and the integrator email (standard HTTP multipart request).

i Note

Integrators can implement both steps in whatever language and platform their systems already use, as both are standard protocols. The examples provided show the raw requests for clarity.

What you need before you start:

1. A CCRS M2M account, provisioned through Washington State's identity platform. You will be issued a client ID and client secret.
2. One or more authorized uploader email addresses (primary and/or backup) registered for the integrator account.
3. The API identifier (audience) and token endpoint for the environment you are connecting to - both are listed under Request Requirements below.

Request Requirements

Every file submission must include the following.

Identity platform token endpoints

Pre-Production

Token endpoint: <https://test-login.wa.gov/oauth/token>

API identifier: lcb.ccrs.api.pre

Production

Token endpoint: <https://www.login.wa.gov/oauth/token>

API identifier: lcb.ccrs.api.prod

CCRS upload endpoint

POST <https://<ccrs-environment-host>/api/v1/upload>

ccrs-environment-host:

(Pre-Production) precannabisreporting.lcb.wa.gov

(Production) cannabisreporting.lcb.wa.gov

Example:

(Pre-Production) <https://precannabisreporting.lcb.wa.gov/api/v1/upload>

(Production) <https://cannabisreporting.lcb.wa.gov/api/v1/upload>

Required headers

Header	Value
Authorization	Bearer <access token> - the token from step 1
X-Uploader-Email	<primary or backup integrator email> - must be registered for your integrator

File requirements

- **Format:** CSV, following the CCRS file specifications for the data type being submitted.
- **File name:** must include integrator identifier per the CCRS file naming convention. Each file's name identifies which integrator and data type it represents.
- **One integrator per submission:** files in a single request should belong to your integrator account. Please refer to the [CCRS FAQ page](#) for more information.

Connecting Examples

⚠ Important Note

The examples below are for reference only. They show a standard OAuth 2.0 token request and a standard HTTPS upload request, which is the underlying protocol rather than a supported CCRS implementation. Integrators are responsible for designing, building, testing, securing, and supporting their own integration. LCB does not provide development support.

The examples use curl to show the raw HTTP requests in a way that does not depend on any one language. Use them to understand the format, then build the same requests in your own system.

Step 1 - Get an access token

Send your client ID, client secret, and API identifier to the identity platform's token endpoint using the OAuth 2.0 client-credentials grant. You receive a bearer token valid for a limited time.

The example below uses the pre-production configuration values. Replace only YOUR_CLIENT_ID and YOUR_CLIENT_SECRET with the credentials issued to you. The audience value is required and must be the API identifier for the environment you are submitting to. A token requested with the wrong audience is still issued by the identity platform, but CCRS rejects it at upload.

```
curl --request POST \
  --url https://test-login.wa.gov/oauth/token \
  --header "content-type: application/json" \
  --data '{
    "client_id": "YOUR_CLIENT_ID",
    "client_secret": "YOUR_CLIENT_SECRET",
    "audience": "lcb.ccrs.api.pre",
    "grant_type": "client_credentials"
  }'
```

A successful response includes an access token:

```
{
  "access_token": "eyJhbGciOi...<token>...",
  "token_type": "Bearer",
  "expires_in": 86400
}
```

Tokens expire 24 hours after they are issued. Request a new token when needed rather than hard-coding one. Cache and reuse a token until close to its expiry rather than requesting one per file.

Step 2 - Submit a file

Send your file to the CCRS API endpoint, including the bearer token from Step 1 and the uploader email header.

This example sends a single file as multipart form data. Replace the host with the correct CCRS environment, the token with the one from Step 1, the email with an authorized uploader email, and the file with your correctly named CCRS data file.

```
curl --request POST \  
  --url https://<ccrs-environment-host>/api/v1/upload \  
  --header "Authorization: Bearer eyJhbGciOi...<token>..." \  
  --header "X-Uploader-Email: uploader@example.com" \  
  --form "files=@Area_YOURIDENTIFIER_20260101120000.csv"
```

Submitting more than one file in a single request

To send several files at once, repeat the `--form "files=@..."` argument once per file. All files in a single request must belong to your integrator account.

```
curl --request POST \  
  --url https://precannabisreporting.lcb.wa.gov/api/v1/upload \  
  --header "Authorization: Bearer eyJhbGciOi...<token>..." \  
  --header "X-Uploader-Email: uploader@example.com" \  
  --form "files=@Area_YOURIDENTIFIER_20260101120000.csv" \  
  --form "files=@Strain_YOURIDENTIFIER_20260101120001.csv" \  
  --form "files=@Product_YOURIDENTIFIER_20260101120002.csv"
```

Please note: a submission is validated as a set. If any file fails validation, the whole request is rejected and none of the files are accepted. The response lists each problem. Correct the listed files and resubmit. A partial batch is never left half-processed.

A note on the standard

The token request follows the standard OAuth 2.0 client-credentials flow. For authoritative details, refer to the OAuth 2.0 specification and the identity platform's documentation provided with your credentials. CCRS follows the standard and does not define a proprietary authentication method.

Responses and What to Expect

The API returns standard HTTP status codes. Common cases:

Result	Meaning
200 / success	The file was accepted for processing.
400	The request was malformed or missing required information (e.g. no file, or a missing header).
401	Authentication failed - missing, invalid or expired token; a token requested with the wrong API identifier (audience); or an uploader email that is not authorized for your integrator.
500	An unexpected error occurred and your submission was not accepted. Retry; if the problem persists, contact LCB.
Rejected file name	A duplicate file name, or a name that doesn't meet the CCRS naming convention.

Responses:

Success (200)

```
{
  "success": true,
  "message": "Your submission was received at <date/time> Pacific Time",
  "warnings": []
}
```

Failure - single reason (400, 401, 500)

```
{
  "success": false,
  "error": "Missing required header: X-Uploader-Email"
}
```

Failure - one or more files rejected during validation (400)

```
{
  "success": false,
  "errors": [ "<reason>", "<reason>" ]
}
```

Note the two failure shapes: a single "error" string for a rejected request, and an "errors" array when individual files fail validation. Handle both.

Not every 401 includes a JSON body. A missing, malformed, expired or wrong-audience token is rejected before the request reaches CCRS. That response has an empty body and a WWW-Authenticate header. Treat any 401 as an authentication failure rather than relying on a JSON error field being present. A request larger than the server limit is likewise rejected before CCRS sees it and returns an HTML error page rather than JSON.

Acceptance means the file was received for processing, not that every row passed validation. Downstream processing validates the data. Please consult the CCRS file specifications to ensure your data is correct.

Testing and Support

- Integrators are responsible for building and testing their own connection and submissions.
- CCRS does not provide integration development or debugging support. The requirements and illustrative examples here are provided so you can build and verify your own systems.

For account provisioning or reporting-requirement questions, please send in a ticket to the LCB ITS Service Desk (servicedesk@lcb.wa.gov).

Common Issues

Where do we get our credentials?

M2M accounts are provisioned per the process under “What you need before you start.” Please send in a ticket to Service Desk by emailing servicedesk@lcb.wa.gov.

We can't get an access token.

Confirm your client ID, client secret, and API identifier (audience) are correct and that you're using the standard OAuth 2.0 client-credentials request against the correct token endpoint. Because integrators build and operate their own software, troubleshooting your implementation is your responsibility.

Our submissions are rejected.

Confirm your token is valid and current, the uploader email header is present and authorized, and your files follow the CCRS naming and format specifications.

Glossary

Term	Meaning
M2M account	A machine-to-machine account your software uses to connect to the CCRS API without a person signing in.
Client ID / secret	The credentials issued with your M2M account, used to authenticate.
Access token (bearer token)	A temporary credential your software obtains and presents when submitting files.
OAuth 2.0 client credentials	The standard protocol flow used to obtain an access token from a client ID + secret.
Audience (API identifier)	The identifier of the CCRS API for which your token is issued.
Uploader email	An authorized email, sent in a request header, on whose behalf a submission is made.
Integrator identifier	A code identifying your integrator, included in each file's name per CCRS naming rules.