

eCoC — REST API Reference

Contact: ecocdataenquiries@vca.gov.uk **Version:** 1.2 (released 2026/07/28)

Overview

This API allows the submission of eCoC data records to the VCA eCoC system. It also allows the consumer to query the async processing status of previous submissions.

The **VCA Ingest API** accepts **InitialVehicleInformation (IVI) v2** XML documents and queues them for downstream processing. Successful submissions return **202 Accepted**. Validation errors and header issues return structured errors and appropriate HTTP status codes. All traffic is served via **Azure Front Door**.

- **Style:** REST over HTTPS
- **Primary media type:** `application/xml`
- **Auth:** OAuth 2.0 (Client Credentials)

Before you start

To get started, you will be provided with the following credentials/information:

Parameter	Description
tenant	VCA Tenant ID for auth.
manufacturer-client-id	Your organisation's GUID used when obtaining auth tokens for the API.
manufacturer-secret	Your organisation's secret used when obtaining auth tokens for the API.
scope	OAuth scope for token acquisition. This will differ between our test and production environments.
manufacturer-id	Your registered manufacturer ID (GUID), this ties your submissions to your organisation to enable you to view statuses in our Portal.
contact-id	Your registered contact ID (GUID), this ties your submissions to a specific user identity which you can then log into the Portal with to view your submissions' statuses.

Quick Start

Environment Values

Use the environment-specific scope and host that match the target environment:

Environment	Scope	Host
Preprod	<code>api://7d84c5ec-ed94-4082-b329-9a7e0b92034f/.default</code>	<code>https://ecoc-api-preprod.vehicle-certification-agency.gov.uk</code>
Prod	<code>api://7f347c17-ecb9-4abc-b595-1040951a27ad/.default</code>	<code>https://ecoc-api.vehicle-certification-agency.gov.uk</code>

The examples below use Preprod values. For Prod, replace the scope and host with the values from the table above.

Step 1 — Get a token using your Client Credentials

- tenant: 37e4cd23-987e-4c94-a17d-c1ea08f6ebd9
- client_id: {{manufacturer-client-id}}
- client_secret: {{manufacturer-secret}}
- scope: use the environment-specific value from the table above. Example: api://7d84c5ec-ed94-4082-b329-9a7e0b92034f/.default **[Preprod environment]**

Requesting a Token

To obtain an access token, send a POST request to the token endpoint:

`https://login.microsoftonline.com/37e4cd23-987e-4c94-a17d-c1ea08f6ebd9/oauth2/v2.0/token`

Headers

- Content-Type: application/x-www-form-urlencoded

Body (form-data) **[Preprod environment]**

- client_id: {{manufacturer-client-id}}
- client_secret: {{manufacturer-secret}}
- grant_type: client_credentials
- scope: api://7d84c5ec-ed94-4082-b329-9a7e0b92034f/.default

Example cURL **[Preprod environment]**

```
curl -X POST "https://login.microsoftonline.com/37e4cd23-987e-4c94-a17d-c1ea08f6ebd9/oauth2/v2.0/token" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "client_id={{manufacturer-client-id}}" \
-d "client_secret={{manufacturer-secret}}" \
-d "grant_type=client_credentials" \
-d "scope=api://7d84c5ec-ed94-4082-b329-9a7e0b92034f/.default"
```

Use the token in Authorization: Bearer <token> for all requests.

Step 2 — POST your IVI XML to the ingest endpoint

POST to the ingest endpoint with the required headers. **[Preprod environment]**

```
curl -X POST "https://ecoc-api-preprod.vehicle-certification-agency.gov.uk/api/ingest" \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/xml" \
-H "x-manufacturer-id: {{manufacturer-id}}" \
-H "x-contact-id: {{contact-id}}" \
-H "x-correlation-id: $(uuidgen)" \
--data-binary @ivi.xml -i
```

Expected: 202 Accepted on success. Validation issues return 4xx with details (see Errors).

Base URL

All traffic must target the hostname for the selected environment.

Environment	Host
-------------	------

Preprod	https://ecoc-api-preprod.vehicle-certification-agency.gov.uk
Prod	https://ecoc-api.vehicle-certification-agency.gov.uk

Authentication

Scheme: OAuth 2.0 — Client Credentials

Audience/Scope: use the environment-specific scope for the target environment:

Environment	Scope
Preprod	api://7d84c5ec-ed94-4082-b329-9a7e0b92034f/.default
Prod	api://7f347c17-ecb9-4abc-b595-1040951a27ad/.default

- **Header:** Authorization: Bearer <token>

In addition to Bearer auth, the API requires identification headers (x-manufacturer-id, x-contact-id).

Required Headers

Header	Required	Value / Format	Notes
Authorization	Yes	Bearer <access_token>	OAuth2
Accept	Yes	application/json, text/html	Accept JSON / HTML responses
Content-Type	Yes	application/xml	XML payload only
x-manufacturer-id	Yes	GUID	manufacturer-id (provided to you on registration)
x-contact-id	Yes	GUID	contact-id (provided to you on registration)
x-correlation-id	Recommended	GUID	For traceability (generated by you, or the system will generate one and provide it in the response if this header is not included)

Endpoint Summary

1) Ingest IVI document

POST /api/ingest

Body: IVI v2 XML (application/xml).

Sample minimal body (truncated)

```
<?xml version="1.0" encoding="UTF-8"?>
<InitialVehicleInformation xmlns="http://eu.ereg.initialvehicleinformation.v2"
    xmlns:sig="http://www.w3.org/2000/09/xmldsig#">
  <Header>
    <IviReferenceId>...</IviReferenceId>
    <IviVersionNumberXsd>2.0</IviVersionNumberXsd>
    <IviVersionNumber>1</IviVersionNumber>
    <IviVersionDateTime>2025-12-12T10:30:00Z</IviVersionDateTime>
    <IntendedCountryRegistration>D</IntendedCountryRegistration>
    <DesignatedTypeApprovalCountry>e1</DesignatedTypeApprovalCountry>
  </Header>
  <CocDataGroup>
    <Type>...</Type>
    <Variant>...</Variant>
    <Version>...</Version>
    <TypeApprovalNumber>...</TypeApprovalNumber>
    <TypeApprovalIssueDate>...</TypeApprovalIssueDate>
    <VehicleIdentificationNumber>{{vin}}</VehicleIdentificationNumber>
    <!-- ...other groups/tables... -->
  </CocDataGroup>
</InitialVehicleInformation>
```

cURL [Preprod environment]

```
curl -s -X POST "https://ecoc-api-preprod.vehicle-certification-agency.gov.uk/api/ingest" \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/xml" \
-H "x-manufacturer-id: {{manufacturer-id}}" \
-H "x-contact-id: {{contact-id}}" \
--data-binary @ivi.xml -i
```

Responses

- **202 Accepted** — submission queued/accepted.
- **204 No Content** — missing authorization headers.
- **400 Bad Request** — missing headers, empty body, or malformed XML.
- **403 Forbidden** — invalid auth token. Or payload exceeds size limit (max 128 kb).
- **409 Conflict** — duplicate submission.
- **413 Payload Too Large** — payload exceeds size limit of 1 mb.
- **415 Unsupported Media Type** — wrong Content-Type (must be application/xml).
- **422 Unprocessable Entity** — XML is syntactically valid but fails domain validation; response body is JSON with error and message fields.
- **429 Too Many Requests** — the VCA system implements service protection limits, if our system is currently receiving too many requests across all clients, you may receive this response; retry with backoff.

Response 202 (example)

```
{
  "status": "accepted",
  "correlationId": "b1f8e13e-24ab-4cf8-9877-60ddd7375da6",
  "location": "2026/03/04/b1f8e13e-24ab-4cf8-9877-60ddd7375da6",
  "iviReferenceId": "59c6696a-6e55-4539-80f1-8917986cf37f"
}
```

Error Shape (422 Unprocessable Entity)

```
{
  "error": "validation_failed",
```

```
"message": "The actual length is greater than the MaxLength value",  
"details": [ { "field": "...", "issue": "..." } ]  
}
```

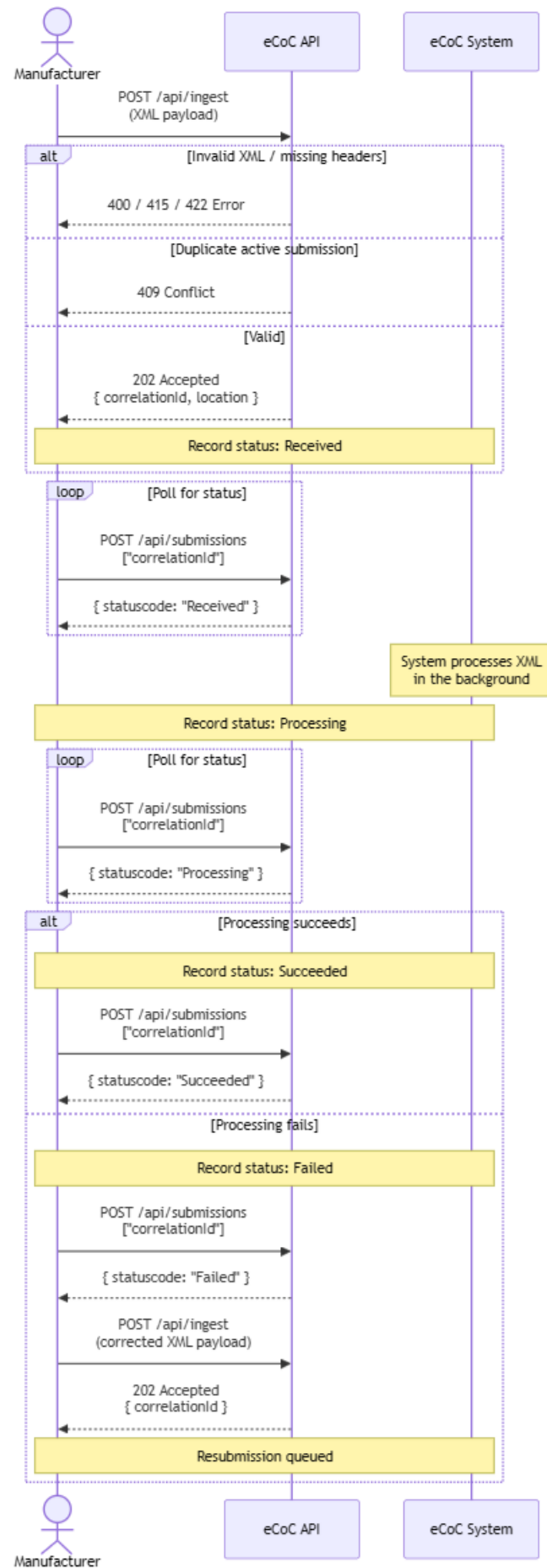
2) Submission status lookup (batch)

POST /api/submissions

Submit a JSON array of correlation IDs to retrieve status/metadata for multiple accepted submissions. Note that this endpoint is for retrieving metadata/status only, not the original IVI document. Note also that this will only return results for correlation IDs that were accepted (i.e. returned 202 Accepted on submission); any invalid/unknown correlation IDs will be ignored in the response.

Submission Status Flow

The diagram below summarises the lifecycle of a submission through the system:



Headers

- Authorization: Bearer <token>
- Accept: application/json, text/html
- Content-Type: application/json

Request Body

```
[
  "{{correlation_id_1}}",
  "{{correlation_id_2}}",
  "{{correlation_id_3}}"
]
```

Response 200 (example)

```
{
  "{{correlation_id_1}}": {
    "ceox_iversionnumber": 1,
    "ceox_typeapprovalnumber": "...",
    "ceox_vehiclecategory": "M1",
    "ceox_vehicleidentificationnumber": "...",
    "statuscode": 200,
    "createdon": "2026-03-03T17:11:56Z"
  }
}
```

Errors

- **400 Bad Request** — invalid/empty JSON.
- **403 Forbidden** — missing/invalid auth token. Or payload exceeds size limit (max 128 kb).
- **415 Unsupported Media Type** — wrong Content-Type.
- **429 Too Many Requests** — the VCA system implements service protection limits, if our system is currently receiving too many requests across all clients, you may receive this response; retry with backoff.
- **500 Internal Server Error** — unexpected processing failure.

Backoff and Retries

When calling this API, you may occasionally receive 429 Too Many Requests if traffic exceeds allowed limits. Because the endpoint does not issue Retry-After headers, clients must implement their own backoff behaviour to ensure stability and prevent overwhelming the service.

1) Retry Only When Appropriate

Clients should retry only when receiving one of the following:

- 503 Service Unavailable
- 504 Gateway Timeout
- Network timeouts

Do **not** retry:

- 4xx validation errors
- Authentication failures
- Any 403 response

2) Use Exponential Backoff with Jitter

Apply controlled retry spacing:

- Initial delay: **1 second**
- Backoff pattern: $\text{delay} = \text{baseDelay} \times 2^{\text{attempt}}$
- Add jitter: $\pm 20\text{--}30\%$
- Maximum delay cap: **32 seconds**
- Maximum retry attempts: **5**

Attempt	Base Delay	Backoff with Jitter
1	1s	0.8–1.3s
2	2s	1.6–2.6s
3	4s	3.2–5.2s
4	8s	6.4–10.4s
5	16s	12.8–20.8s

3) Avoid Synchronous Burst Retries

- Retries from different requests must be scheduled independently.
- Do not retry all failed calls at the same moment — this causes traffic spikes and further throttling.

Observability

- Provide x-correlation-id for end-to-end tracing (echoed in logs and responses where applicable).
- Front Door may add diagnostic headers such as X-Azure-Ref for debugging any issues.