

ELD NextGen Policy API

Outline Specification

Contents

1	Purpose of this Document	2
2	Conventions	2
2.1	Authentication	2
2.2	Insurer Selection	3
2.3	Enumerations.....	3
2.4	Common response codes	3
2.5	Error response shape.....	3
2.6	Policy Flow	4
3	Endpoints	4
3.1	UPSERT Policies (Direct)	4
3.2	DELETE Policies (Direct)	5
3.3	UPLOAD File	6
3.4	File Format.....	8
3.5	GET Submission Status.....	9
3.6	Test Endpoints	11
4	Object Schemas.....	11
5	Validation Rules	11
5.1	Overview.....	11
5.2	Policy-level fields	11
5.3	Employer-level fields.....	12
5.4	Duplicate detection.....	12
5.5	File format validation	12
5.6	DELETE validation	12

1 Purpose of this Document

This document is an early view of the ELD Policy Data API. It is provided to help planning and will be supported by full API documentation and test endpoints. These will be released in phases from July 2026.

It covers authentication, endpoint contracts, request/response schemas and validation rules.

Still to be confirmed:

The following elements are still to be confirmed and will be shared in the full API documentation.

- Full Open API Specification and example of worked payload including multiple employers, ERNs and successful response
- Base URL for test and production endpoints
- API Credentials and token authentication
- Status Polling cadence and back-off strategy
- Retry behaviour and idempotency guidance
- Guidance on rate limiting
- Guidance on payload and file sizes
- Full error catalogue and error responses

For business context and a narrative summary of what is changing, see: **ELD NextGen Policy Management Summary of Changes document**.

NOTE:

All information provided in this document is subject to change before final release of the API.

2 Conventions

2.1 Authentication

The API uses **Microsoft Entra** authentication. Suppliers will be able to generate credentials to authenticate API requests. The expected mechanism is a **client credentials flow** and the exact details will be confirmed in the next version of the document.

2.2 Insurer Selection

The supplier will be inferred from the credentials used. The insurer will be provided explicitly in the path of the request and validated against the credentials used to provide additional assurance that policies are being uploaded against the correct insurer.

2.3 Enumerations

All enum values are **case-insensitive** — the API accepts any casing and normalises internally.

2.4 Common response codes

All validation errors, including duplicate policy records, invalid dates, invalid ERNs/CRNs and missing required fields, are returned as `400 Bad Request` with a structured list of validation errors. `409 Conflict` and `422 Unprocessable Entity` are not used for policy validation errors in this version of the API.

These response codes apply to every endpoint, in addition to the per-endpoint responses listed below.

Code	Meaning
400 Bad Request	The request is malformed or invalid
401 Unauthenticated	No valid authentication credentials provided.
403 Forbidden	Authenticated but insufficient role or permissions for this operation.
404 Not Found	Supplier or insurer not found, or insufficient access to the organisation(s).

2.5 Error response shape

Every validation error response contains a list of errors, each pointing to a specific field or item (e.g. it may be an entire employer or policy) via a dot/bracket JSON-pointer-style notation.

```
{
  "path": "policies[0].employers[1].name",
  "description": "Employer name is required.",
  "policyNumber": "1",
  "supplementaryPolicyReference": "SPR",
  "coverStartDate": "2020-01-01",
  "coverEndDate": "2025-12-31"
}
```

2.6 Policy Flow

Direct API submissions are validated synchronously. If validation fails, no policies in the direct submission are accepted or queued. Where validation passes, the API response confirms that the submission has been accepted and queued for ingestion.

File-based submissions are accepted for processing after the file upload has completed. Validation and ingestion then take place asynchronously. Valid policies are ingested individually and invalid policies are rejected individually, with validation errors available through the status endpoint and the ELD policy management area.

Acceptance or queuing does not mean the policy change will be immediately visible in all ELD features. There may be a short delay before changes appear in policy search, reporting, extracts or downstream ELD functions.

Policies must be supplied in full. Partial updates are not supported. Where a policy with the same policy key already exists, the newly accepted policy submission replaces the previous version of that policy once ingestion is complete.

3 Endpoints

3.1 UPSERT Policies (Direct)

POST /insurers/<insurerId>/policies

Submit one or more policies for upsert. Validation is synchronous.

Behaviour

All-or-nothing. If any policy in the submission fails validation, the entire submission is rejected and no policies are ingested.

Request body:

Array of PolicyUpsert objects (see schema in section 4).

```
[
  {
    "policyNumber": "POL001",
    "supplementaryPolicyReference": "SPR-ABC",
    "coverStartDate": "2015-01-01",
    "coverEndDate": "2023-12-31",
    "policyholderName": "Example Policyholder Ltd",
    "employers": [
      {
```

```

    "name": "Example Manufacturing Ltd",
    "erns": ["123/AB12345", "456/CD6789"],
    "crn": "12345678",
    "singleLineAddress": "Example Street, Leeds, LS1 1AB",
    "adjustedCoverStartDate": "2015-06-01",
    "adjustedCoverEndDate": "2022-06-30"
  },
  {
    "name": "Example Services Ltd",
    "erns": ["789/EF11111"],
    "multilineAddress": {
      "addressLine1": "Example Park",
      "addressLine2": "Example Quarter",
      "cityTown": "Manchester",
      "county": "Greater Manchester",
      "country": "England",
      "postCode": "M1 1AA"
    },
    "adjustedCoverStartDate": "2016-03-01",
    "adjustedCoverEndDate": "2021-09-30"
  }
]
}
]

```

Responses:

Code	Meaning
200 OK	All policies accepted and queued for ingestion.
400 Bad Request	One or more validation errors. No policies have been accepted. Response body contains all errors as an array of ValidationError objects..

Note

A 200 OK response confirms that the submission has passed validation and has been queued for ingestion. It does not guarantee that the policy changes are immediately visible in ELD search, reporting or downstream outputs. Use the status endpoint to confirm processing completion.

3.2 DELETE Policies (Direct)

DELETE /insurers/<insurerId>/policies

Submit one or more policies for deletion. Only key fields are required. Submitting the policy key will delete the whole policy represented by that key, including all employers.

Behaviour

All-or-nothing. If any entry fails validation, the entire submission is rejected.

Request body:

```
{
  "policyNumber": "string",
  "supplementaryPolicyReference": "string (optional)",
  "coverStartDate": "YYYY-MM-DD"
}
```

Responses: As UPSERT (Direct).

3.3 UPLOAD File

POST /insurers/<insurerId>/policies/file

Initiate a file-based policy upload. Returns an upload ID and a secure upload URL. Uploading the file to the URL will queue the file for ingestion. At this point the file will be made read-only and should not be attempted to be modified once uploaded.

Behaviour

Valid policies are ingested individually; invalid policies are rejected individually. There is no minimum pass rate.

DELETE file submissions follow the same per-record processing as UPSERT files — valid deletions are processed; invalid entries are rejected individually with errors available in the status response

Request body:

```
{
  "fileType": "JSON | TEXT",
  "fileName": "string",
  "operationType": "UPSERT | DELETE"
```

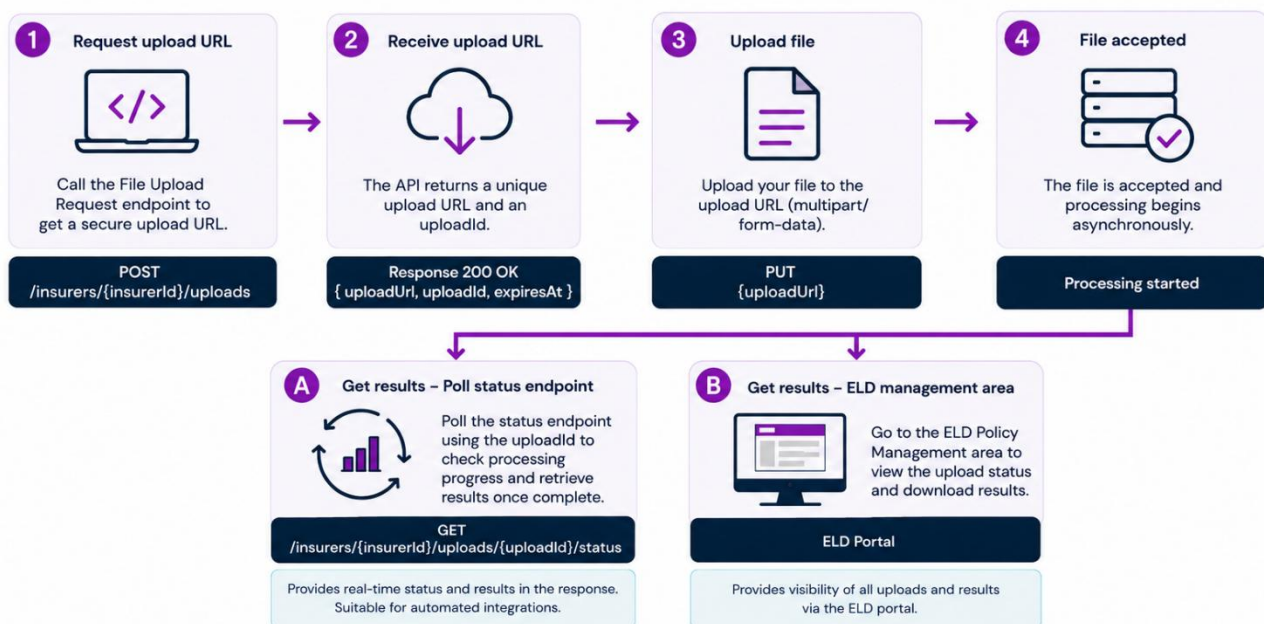
```
}
```

Field	Description
uploadId	Identifier for the submission.
timestamp	When the submission was initiated (used for ordering).
operationType	UPSERT or DELETE.
status	One of the status values listed below.

Response (200 OK):

```
{
  "uploadId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "uploadUrl": "https://...",
  "expiresAt": "2026-04-21T17:00:00Z",
  "timestamp": "2026-04-20T16:00:00Z"
}
```

Code	Meaning
200 OK	Upload initiated — use uploadId to track status.
400 Bad Request	Invalid file type or request.



Three-step flow

1. POST to /file → receive uploadId + uploadUrl.
2. Upload the file directly to uploadUrl — processing will start automatically after the upload completes.
3. Poll uploadId until status is one of the terminal states: COMPLETE, REJECTED, or SYSTEM_ERROR.

3.4 File Format

The file format should be TEXT pipe-delimited format. The column order is detailed below. The ERN is placed last as related identifier field, so that all preceding fields repeat naturally across multiple ERN rows.

#	Column
1	PolicyNumber
2	SupplementaryPolicyReference
3	CoverStartDate
4	CoverEndDate
5	PolicyholderName
6	FirstClaimSettlementDate
7	EmployerName
8	SingleLineAddress
9	AddressLine1
10	AddressLine2
11	CityTown
12	County
13	Country
14	PostCode
15	AdjustedCoverStartDate
16	AdjustedCoverEndDate
17	CRN
18	ERN

Fields Containing Pipe Characters

The pipe character (|) is used as the field separator in the text file format. If a field value itself contains a pipe character, the whole field must be enclosed in double quotes. This prevents the pipe character from being interpreted as a new column (following standard formatting in Excel)

For example, if an employer name is:

```
Employer | Trading Ltd
```

it must be supplied as:

```
"Employer | Trading Ltd"
```

Example row:

```
POL12345|SUPP01|2026-01-01|2026-12-31|Policyholder Ltd||"Employer | Trading Ltd"||1 High Street|Business Park|London||UK|SW1A 1AA|||123/AB12345
```

If a field contains double quotes, each double quote must be doubled. For example:

```
Employer "North" Ltd
```

must be supplied as:

```
"Employer ""North"" Ltd"
```

3.5 GET Submission Status

GET /insurers/<insurerId>/policies/status/{uploadId}

The status response returns validation errors for rejected policies and records within a File submission. Each validation error identifies the location of the issue using a structured path, together with a human-readable description of the error and providing sufficient detail for the submitting organisation to identify the affected record(s) in the original file. This may include (where appropriate) the policy key, employer name, policy index, employer index, field path and error description. Where applicable, row numbers may also be provided for pipe-delimited text files.

Validation results can be viewed through the API status response and the ELD policy management area.

Validation error detail

The system does not issue a separate return file after processing. For file-based submissions, validation results are retrieved using the status endpoint or viewed in the ELD policy management area.

The status response includes counts of total, accepted and rejected policies, together with validation errors for rejected records. Each validation error identifies the affected

item using a structured path and a human-readable error description. For pipe-delimited text files, the response should also identify the affected row where this is available, so that insurers can locate and correct the source record.

Response (200 OK):

```
{
  "id": 0,
  "uploadId": "<GUID>",
  "timestamp": "<UTC ISO timestamp>",
  "uploadType": "WEB/SYNC/API_FILE/API_DIRECT",
  "fileName": "",
  "operationType": "UPSERT/DELETE",
  "status": "WAITING_FOR_FILE/QUEUED/PROCESSING/COMPLETE/REJECTED/SYSTEM_ERROR",
  "fileRejectionReason": "<string>",
  "totalPolicies": 1,
  "validPolicies": 1,
  "rejectedPolicies": 1,
  "validationErrors": [
    {
      "description": "Error description",
      "path": "policy[0].employers[0].name"
    }
  ],
  "processingStartedAt": "<UTC ISO timestamp>",
  "validationCompletedAt": "<UTC ISO timestamp>",
  "processingCompletedAt": "<UTC ISO timestamp>"
}
```

Status values:

Status	Description
WAITING_FOR_FILE	Upload record created; waiting for file to be uploaded to the SAS URL.
QUEUED	File received; queued for processing.
PROCESSING	Validation and ingestion in progress.
COMPLETE	Processing complete.
REJECTED	File rejected (e.g. invalid file format) — no policies processed.
SYSTEM_ERROR	Unexpected internal error — contact support.

Code	Meaning
200 OK	Status returned.
404 Not Found	Upload ID not found, or insufficient access (e.g. owned by a different supplier).

3.6 Test Endpoints

Full test versions of each of the core endpoints will be made available via the same API. They will behave identically to the live endpoints, including validation, but do not ingest data for use in the ELD.

These endpoints are intended to support development and integration testing. The endpoints will use separate credentials so as not to risk accidental inclusion of test data.

Test endpoint credentials will be available via the ELD Portal and full documentation available via the developer portal.

4 Object Schemas

Full object schemas for all end points will be provided as part of the full API documentation within the Developer Portal.

5 Validation Rules

5.1 Overview

All policy submissions are validated. For direct submissions, all errors across all policies are collected and returned together in a single response. For file submissions, errors are available via the status endpoint (and policy management area of the portal) after validation completes.

All unicode characters will be supported going forwards.

Note

The validation rules below are indicative of what will be enforced. The final validation rule set will be confirmed when the full API specification is released.

5.2 Policy-level fields

Field	Required	Rules
policyNumber	Required	Must not start or end with whitespace.
supplementaryPolicyReference	Optional	If provided, must not start or end with whitespace.

coverStartDate	Required	ISO 8601 YYYY-MM-DD; must be after 01/01/1900; must be before coverEndDate.
coverEndDate	Required	ISO 8601 YYYY-MM-DD; after 01/01/1900; after coverStartDate.
policyholderName	Required	—
firstClaimSettlementDate	Optional	ISO 8601 YYYY-MM-DD; only valid if coverEndDate is before the original ELD go-live date (2011-04-01); value must be on or after the original ELD go-live date (2011-04-01).
employers	Required	At least one employer must be present.

5.3 Employer-level fields

Field	Required	Rules
employers[].name	Required	—
employers[].erns	Optional	Array; each ERN must be in the correct format.
employers[].crn	Optional	Must be in the correct format.
employers[].singleLineAddress	Required (one of)	Exactly one of singleLineAddress or multilineAddress must be provided.
employers[].multilineAddress	Required (one of)	Providing any subfield counts as supplying a multiline address.
employers[].multilineAddress.postCode	Optional	Must be in the correct format.
employers[].adjustedCoverStartDate	Optional	Used when a specific employer's cover started after the overall policy. ISO 8601 YYYY-MM-DD; after 01/01/1900; after coverStartDate; before adjustedCoverEndDate if both provided, otherwise before policy coverEndDate.
employers[].adjustedCoverEndDate	Optional	Used when a specific employer's cover ended before the overall policy. ISO 8601 YYYY-MM-DD; before coverEndDate; after adjustedCoverStartDate if both provided, otherwise after policy coverStartDate.

5.4 Duplicate detection

Rule	Behaviour
Duplicate policy within submission	Two policies with the same policyNumber, supplementaryPolicyReference and coverStartDate— the first is processed; subsequent duplicates are rejected.

5.5 File format validation

Format	Check
JSON	File must be valid JSON; if not, the entire file is immediately rejected.
Pipe-delimited text	File must be parseable as pipe-delimited; header row must be present and match expected columns exactly; each data row must have the correct column count. Invalid format results in immediate rejection of the whole file. Fields containing the pipe character must be enclosed in double quotes following Excel CSV conventions; embedded double quotes must be doubled ("").

5.6 DELETE validation

DELETE submissions follow the same validation logic as an UPSERT.

Submission route	Validation timing	Failure behaviour
Direct DELETE API	Synchronous	All-or-nothing. If one entry fails, none are processed.
File-based DELETE	Asynchronous	Per-record. Valid delete records are processed; invalid delete records are rejected individually.
Invalid DELETE file format	Immediate file-level rejection	No records are processed.

NOTE: When a Delete submission is processed if the format is valid it will be processed. There is no response on the number of records deleted. This will need to be validated separately via your policy reports.

Policy-level fields (DELETE)

Field	Required	Rules
policyNumber	Required	Must not start or end with whitespace.
supplementaryPolicyReference	Optional	If provided, must not start or end with whitespace.
coverStartDate	Required	ISO 8601 YYYY-MM-DD; after 01/01/1900; before coverEndDate.

(No employer-level validation applies — the DELETE endpoint accepts key fields only.)

Duplicate detection (DELETE)

Rule	Behaviour
Duplicate policy within submission	Two entries sharing the same policyNumber, supplementaryPolicyReference and coverStartDate— the first is processed; subsequent duplicates are rejected.