

Test Ingestion APIs Specification

Contents

Direct Policy Upload – Test endpoint (no persistence)	3
Direct Policy Deletions – Test endpoint (no persistence)	22
Get Test Policy File Upload Status	37
Health Check	56
Request a Test Policy File Upload	64

Please note: This document version was issued on 10th August 2026 and will be superseded by live technical specifications hosted on the ELD Developer Portal from 24th August 2026.

Direct Policy Upload – Test endpoint (no persistence)

Validates a JSON array of policies without persisting any data. Returns an uploadId and timestamp on success. All validation errors are collected and returned together.

This endpoint is not idempotent.

Endpoint:

POST <https://test-api.eltohub.org.uk/v1/test/insurers/{insurerId}/policies>

Request

Request parameters

Name	In	Required	Type	Description
insurerId	template	true	integer	The ID of the insurer organisation

Request body

UpsertPoliciesRequest

Name	Required	Type	Description
[]	true	PolicyInput[]	

UpsertPoliciesRequest (json)

```
{
  "minItems": 1,
  "type": "array",
  "items": {
    "$ref": "#/components/schemas/PolicyInput"
  }
}
```

default (json)

```
[{
  "policyNumber": "string",
  "supplementaryPolicyReference": "string",
  "coverStartDate": "string",
  "coverEndDate": "string",
  "policyholderName": "string",
  "firstClaimSettlementDate": "string",
  "originalInsurerName": "string",
  "insurerRef": "string",
  "employers": [{
    "name": "string",
```

```

    "erns": ["string"],
    "crn": "string",
    "singleLineAddress": "string",
    "multilineAddress": {
      "line1": "string",
      "line2": "string",
      "line3": "string",
      "town": "string",
      "county": "string",
      "postCode": "string"
    },
    "adjustedCoverStartDate": "string",
    "adjustedCoverEndDate": "string"
  }
}

```

Response: 200 OK

Policies passed validation — no data persisted

UpsertPoliciesResponse

Name	Required	Type	Description
uploadId	true	string (uuid)	Unique identifier for this upload
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of when the request was received

UpsertPoliciesResponse (json)

```

{
  "required": [
    "uploadId",
    "timestamp"
  ],
  "type": "object",
  "properties": {
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for this upload",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "UTC ISO 8601 timestamp of when the request was received",
      "format": "date-time"
    }
  }
}

```

default (json)

```
{
  "uploadId": "abc-123",
  "timestamp": "2026-06-23T08:50:07Z"
}
```

Response: 400 Bad Request

Validation failed — all errors returned

ApiResponseResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponseResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        }
      }
    }
  }
}
```

```

    },
    "details": {
      "description": "Additional error details (e.g. validation
issues)",
      "nullable": true
    }
  }
}

```

invalidInsurerId (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "VALIDATION_FAILED",
    "message": "Invalid insurer id provided"
  }
}

```

policyValidationFailed (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "VALIDATION_FAILED",
    "message": "Validation failed",
    "details": {
      "issues": [{
        "path": "policies[0].policyNumber",
        "description": "policyNumber is a required field"
      }, {
        "path": "policies[1].coverStartDate",
        "description": "coverStartDate must be a valid date"
      }
    ]
  }
}

```

Response: 401 Unauthorized

Unauthenticated

ApiResponseResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code

Name	Required	Type	Description
error.message	true	string	Human-readable error message

ApiResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation issues)",
          "nullable": true
        }
      }
    }
  }
}
```

default (json)

```
{
  "requestId": "abc-123",
  "error": {
```

ELTO Employers' Liability
Tracing Office

```

    "code": "UNAUTHORISED",
    "message": "Unauthorised"
  }
}

```

Response: 403 Forbidden

Authenticated credentials do not match the requested insurerId

ApiResponseResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponseResponse (json)

```

{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        }
      }
    }
  }
}

```



```

    },
    "details": {
      "description": "Additional error details (e.g. validation
issues)",
      "nullable": true
    }
  }
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "FORBIDDEN",
    "message": "You do not have permission to access this insurer"
  }
}

```

Response: 413 Request Entity Too Large

Request body exceeds the 2 MB size limit

BodySizeLimitError

Returned when the request body exceeds the 2 MB size limit

Name	Required	Type	Description
code	true	string	
message	true	string	

BodySizeLimitError (json)

```

{
  "required": [
    "code",
    "message"
  ],
  "type": "object",
  "properties": {
    "code": {
      "type": "string",
      "example": "PAYLOAD_TOO_LARGE"
    },
    "message": {
      "type": "string",
      "example": "Request body size 10486494 exceeds the configured
limit of 2097152 bytes."
    }
  }
}

```

```
    },
    "description": "Returned when the request body exceeds the 2 MB size
limit"
}
```

default (json)

```
{
  "code": "PAYLOAD_TOO_LARGE",
  "message": "Request body size 10486494 exceeds the configured limit of
2097152 bytes."
}
```

Response: 500 Internal Server Error

Unexpected internal error

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",

```

```

        "FORBIDDEN",
        "NOT_FOUND",
        "INTERNAL_ERROR"
    ],
    "type": "string",
    "description": "Machine-readable error code"
},
"message": {
    "type": "string",
    "description": "Human-readable error message"
},
"details": {
    "description": "Additional error details (e.g. validation
issues)",
    "nullable": true
}
}
}
}
}
}

```

default (json)

```

{
    "requestId": "abc-123",
    "error": {
        "code": "INTERNAL_ERROR",
        "message": "Internal Server Error"
    }
}

```

Response: 429 Too many requests

Rate limit exceeded

Response headers

Name	Required	Type	Description
Retry-After	false	integer	Number of seconds to wait before retrying
X-RateLimit-Remaining-Minute	false	integer	Remaining requests allowed in the current minute window

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code

Name	Required	Type	Description
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```

default (json)

```
{
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Too many requests. Please retry after the period indicated
in the Retry-After header."
  }
}
```

Definitions

Name	Description
BodySizeLimitError	Returned when the request body exceeds the 2 MB size limit
PolicyInput	
EmployerInput	Exactly one of singleLineAddress or multilineAddress must be provided.

Name	Description
MultilineAddress	Employer address as structured fields. Exactly one of singleLineAddress or multilineAddress must be provided on the parent employer.
UpsertPoliciesRequest	
UpsertPoliciesResponse	
ApiResponseResponse	
RateLimitErrorResponse	

BodySizeLimitError

Returned when the request body exceeds the 2 MB size limit

Name	Required	Type	Description
code	true	string	
message	true	string	

BodySizeLimitError (json)

```
{
  "required": [
    "code",
    "message"
  ],
  "type": "object",
  "properties": {
    "code": {
      "type": "string",
      "example": "PAYLOAD_TOO_LARGE"
    },
    "message": {
      "type": "string",
      "example": "Request body size 10486494 exceeds the configured
limit of 2097152 bytes."
    }
  },
  "description": "Returned when the request body exceeds the 2 MB size
limit"
}
```

PolicyInput

Name	Required	Type	Description
policyNumber	true	string	Unique policy reference number. Must not have leading or trailing whitespace.
supplementaryPolicyReference	false	string	Optional supplementary reference. Must not have leading or trailing whitespace.
coverStartDate	true	string (date)	Policy cover start date (ISO 8601 YYYY-MM-DD). Must be after 1900-01-01.
coverEndDate	true	string (date)	Policy cover end date (ISO 8601 YYYY-MM-DD). Must be after coverStartDate and after 1900-01-01.
policyholderName	true	string	Name of the policyholder.
firstClaimSettlementDate	false	string (date)	First claim settlement date (ISO 8601 YYYY-MM-DD). Only valid for policies with a coverStartDate bef...
originalInsurerName	true	string	Name of the original insurer. Maximum 100 characters.
insurerRef	false	string	Optional insurer reference. Maximum 100 characters.
employers	true	EmployerInput[]	List of employers covered by this policy. At least one employer must be present.

PolicyInput (json)

```
{
  "required": [
```

```
"policyNumber",
"coverStartDate",
"coverEndDate",
"policyholderName",
"originalInsurerName",
"employers"
],
"type": "object",
"properties": {
  "policyNumber": {
    "type": "string",
    "description": "Unique policy reference number. Must not have
leading or trailing whitespace."
  },
  "supplementaryPolicyReference": {
    "type": "string",
    "description": "Optional supplementary reference. Must not have
leading or trailing whitespace."
  },
  "coverStartDate": {
    "type": "string",
    "description": "Policy cover start date (ISO 8601 YYYY-MM-DD).
Must be after 1900-01-01.",
    "format": "date"
  },
  "coverEndDate": {
    "type": "string",
    "description": "Policy cover end date (ISO 8601 YYYY-MM-DD). Must
be after coverStartDate and after 1900-01-01.",
    "format": "date"
  },
  "policyholderName": {
    "type": "string",
    "description": "Name of the policyholder."
  },
  "firstClaimSettlementDate": {
    "type": "string",
    "description": "First claim settlement date (ISO 8601 YYYY-MM-DD).
Only valid for policies with a coverStartDate before 2011-04-01 (the original
ELD go-live date). Must be on or after 2011-04-01.",
    "format": "date"
  },
  "originalInsurerName": {
    "maxLength": 100,
    "type": "string",
    "description": "Name of the original insurer. Maximum 100
characters."
  },
  "insurerRef": {
    "maxLength": 100,
    "type": "string",
    "description": "Optional insurer reference. Maximum 100
characters."
  },
  "employers": {
    "minItems": 1,
```

```

    "type": "array",
    "items": {
      "$ref": "#/components/schemas/EmployerInput"
    },
    "description": "List of employers covered by this policy. At least
one employer must be present."
  }
}

```

EmployerInput

Exactly one of singleLineAddress or multilineAddress must be provided.

Name	Required	Type	Description
name	true	string	Name of the employer.
erns	false	string[]	List of Employer Reference Numbers (ERNs). Each ERN must be a 3-digit tax office number, a slash, then 1–10 alphanumeric characters (e.g. 123/AB12345).
crn	false	string	UK Company Registration Number. 6–8 digits (e.g. 01234567) or 2-letter prefix + 6 digits (e.g. SC123...
singleLineAddress	false	string	Employer address as a single line. Exactly one of singleLineAddress or multilineAddress must be provided. Maximum 1000 characters.
multilineAddress	false	MultilineAddress	Employer address as structured fields. Exactly one of singleLineAddress or multilineAddress must be provided on the parent employer.
adjustedCoverStartDate	false	string (date)	Adjusted cover start date for this employer (ISO 8601 YYYY-MM-DD). Must be after the policy coverStartDate and before adjustedCoverEndDate.
adjustedCoverEndDate	false	string (date)	Adjusted cover end date for this employer (ISO 8601 YYYY-MM-DD). Must be before the policy coverEndDate.

EmployerInput (json)


```
{
  "required": [
    "name"
  ],
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "description": "Name of the employer."
    },
    "erns": {
      "type": "array",
      "items": {
        "pattern": "^[0-9]{3}/[A-Za-z0-9]{1,10}$",
        "type": "string"
      },
      "description": "List of Employer Reference Numbers (ERNs). Each ERN must be a 3-digit tax office number, a slash, then 1-10 alphanumeric characters (e.g. 123/AB12345).",
    },
    "crn": {
      "pattern": "^[A-Za-z]{2}[0-9]{6}|[0-9]{6,8}$",
      "type": "string",
      "description": "UK Company Registration Number. 6-8 digits (e.g. 01234567) or 2-letter prefix + 6 digits (e.g. SC123456).",
    },
    "singleLineAddress": {
      "type": "string",
      "description": "Employer address as a single line. Exactly one of singleLineAddress or multilineAddress must be provided."
    },
    "multilineAddress": {
      "$ref": "#/components/schemas/MultilineAddress"
    },
    "adjustedCoverStartDate": {
      "type": "string",
      "description": "Adjusted cover start date for this employer (ISO 8601 YYYY-MM-DD). Must be after the policy coverStartDate and before adjustedCoverEndDate.",
      "format": "date"
    },
    "adjustedCoverEndDate": {
      "type": "string",
      "description": "Adjusted cover end date for this employer (ISO 8601 YYYY-MM-DD). Must be before the policy coverEndDate.",
      "format": "date"
    }
  },
  "description": "Exactly one of singleLineAddress or multilineAddress must be provided."
}
```

MultilineAddress

Employer address as structured fields. Exactly one of singleLineAddress or multilineAddress must be provided on the parent employer.

Name	Required	Type	Description
line1	false	string	
line2	false	string	
line3	false	string	
town	false	string	
county	false	string	
postCode	false	string	UK postcode.

MultilineAddress (json)

```
{
  "type": "object",
  "properties": {
    "line1": {
      "type": "string"
    },
    "line2": {
      "type": "string"
    },
    "line3": {
      "type": "string"
    },
    "town": {
      "type": "string"
    },
    "county": {
      "type": "string"
    },
    "postCode": {
      "pattern": "^[A-Za-z]{1,2}[0-9][0-9A-Za-z]?\\s?[0-9][A-Za-z]{2}$",
      "type": "string",
      "description": "UK postcode."
    }
  }
},
```

```
"description": "Employer address as structured fields. Exactly one of
singleLineAddress or multilineAddress must be provided on the parent
employer."
}
```

UpsertPoliciesRequest

Name	Required	Type	Description
[]	true	PolicyInput[]	

UpsertPoliciesRequest (json)

```
{
  "minItems": 1,
  "type": "array",
  "items": {
    "$ref": "#/components/schemas/PolicyInput"
  }
}
```

UpsertPoliciesResponse

Name	Required	Type	Description
uploadId	true	string (uuid)	Unique identifier for this upload
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of when the request was received

UpsertPoliciesResponse (json)

```
{
  "required": [
    "uploadId",
    "timestamp"
  ],
  "type": "object",
  "properties": {
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for this upload",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "UTC ISO 8601 timestamp of when the request was
received",
      "format": "date-time"
    }
  }
}
```

ApiResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}
```

```
}
}
}
```

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```

Direct Policy Deletions – Test endpoint (no persistence)

Validates a JSON array of policy deletion keys without persisting any data. Returns an uploadId and timestamp on success. All validation errors are collected and returned together.

This endpoint is not idempotent.

Endpoint:

DELETE <https://test-api.eltohub.org.uk/v1/test/insurers/{insurerId}/policies>

Request

Request parameters

Name	In	Required	Type	Description
insurerId	template	true	integer	The ID of the insurer organisation

Request body

DeletePoliciesRequest

Name	Required	Type	Description
[]	true	DeletePoliciesInput[]	

DeletePoliciesRequest (json)

```
{
  "minItems": 1,
  "type": "array",
  "items": {
    "$ref": "#/components/schemas/DeletePoliciesInput"
  }
}
```

default (json)

```
[{
  "policyNumber": "string",
  "supplementaryPolicyReference": "string",
  "coverStartDate": "string",
  "policyholderName": "string"
}]
```

Response: 200 OK

Policy deletion keys passed validation — no data persisted

DeletePoliciesResponse

Name	Required	Type	Description
uploadId	true	string (uuid)	Unique identifier for this deletion request
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of when the request was received

DeletePoliciesResponse (json)

```
{
  "required": [
    "uploadId",
    "timestamp"
  ],
  "type": "object",
  "properties": {
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for this deletion request",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "UTC ISO 8601 timestamp of when the request was received",
      "format": "date-time"
    }
  }
}
```

default (json)

```
{
  "uploadId": "abc-123",
  "timestamp": "2026-06-23T08:50:07Z"
}
```

Response: 400 Bad Request

Validation failed — all errors returned

ApiResponseResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponseResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}
```


invalidInsurerId (json)

```
{
  "requestId": "abc-123",
  "error": {
    "code": "VALIDATION_FAILED",
    "message": "Invalid insurer id provided"
  }
}
```

policyValidationFailed (json)

```
{
  "requestId": "abc-123",
  "error": {
    "code": "VALIDATION_FAILED",
    "message": "Validation failed",
    "details": {
      "issues": [{
        "path": "policies[0].policyNumber",
        "description": "policyNumber is a required field"
      }, {
        "path": "policies[1].coverStartDate",
        "description": "coverStartDate must be a valid date"
      }]
    }
  }
}
```

Response: 401 Unauthorized

Unauthenticated

ApiResponseResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponseResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
```

```

    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "UNAUTHORISED",
    "message": "Unauthorised"
  }
}

```

Response: 403 Forbidden

Authenticated credentials do not match the requested insurerId

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}
```

default (json)

```
{
  "requestId": "abc-123",
  "error": {
    "code": "FORBIDDEN",
    "message": "You do not have permission to access this insurer"
  }
}
```

Response: 413 Request Entity Too Large

Request body exceeds the 2 MB size limit

BodySizeLimitError

Returned when the request body exceeds the 2 MB size limit

Name	Required	Type	Description
code	true	string	
message	true	string	

BodySizeLimitError (json)

```
{
  "required": [
    "code",
    "message"
  ],
  "type": "object",
  "properties": {
    "code": {
      "type": "string",
      "example": "PAYLOAD_TOO_LARGE"
    },
    "message": {
      "type": "string",
      "example": "Request body size 10486494 exceeds the configured limit of 2097152 bytes."
    }
  },
  "description": "Returned when the request body exceeds the 2 MB size limit"
}
```

default (json)

```
{
  "code": "PAYLOAD_TOO_LARGE",
  "message": "Request body size 10486494 exceeds the configured limit of 2097152 bytes."
}
```

Response: 500 Internal Server Error

Unexpected internal error

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",

```

```

    "nullable": true
  }
}
}
}
}
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "INTERNAL_ERROR",
    "message": "Internal Server Error"
  }
}

```

Response: 429 Too many requests

Rate limit exceeded

Response headers

Name	Required	Type	Description
Retry-After	false	integer	Number of seconds to wait before retrying
X-RateLimit-Remaining-Minute	false	integer	Remaining requests allowed in the current minute window

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```

{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],

```

```

    "type": "object",
    "properties": {
      "code": {
        "type": "string",
        "example": "RATE_LIMIT_EXCEEDED"
      },
      "message": {
        "type": "string",
        "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
      }
    }
  }
}

```

default (json)

```

{
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Too many requests. Please retry after the period indicated
in the Retry-After header."
  }
}

```

Definitions

Name	Description
BodySizeLimitError	Returned when the request body exceeds the 2 MB size limit
DeletePoliciesRequest	
DeletePoliciesInput	
DeletePoliciesResponse	
ApiErrorResponse	
RateLimitErrorResponse	

BodySizeLimitError

Returned when the request body exceeds the 2 MB size limit

Name	Required	Type	Description
code	true	string	
message	true	string	

BodySizeLimitError (json)

```
{
  "required": [
    "code",
    "message"
  ],
  "type": "object",
  "properties": {
    "code": {
      "type": "string",
      "example": "PAYLOAD_TOO_LARGE"
    },
    "message": {
      "type": "string",
      "example": "Request body size 10486494 exceeds the configured
limit of 2097152 bytes."
    }
  },
  "description": "Returned when the request body exceeds the 2 MB size
limit"
}
```

DeletePoliciesRequest

Name	Required	Type	Description
[]	true	DeletePoliciesInput[]	

DeletePoliciesRequest (json)

```
{
  "minItems": 1,
  "type": "array",
  "items": {
    "$ref": "#/components/schemas/DeletePoliciesInput"
  }
}
```

DeletePoliciesInput

Name	Required	Type	Description
policyNumber	true	string	Unique policy reference number. Must not have leading or trailing whitespace.

Name	Required	Type	Description
supplementaryPolicyReference	false	string	Optional supplementary reference. Must not have leading or trailing whitespace.
coverStartDate	true	string (date)	Policy cover start date (ISO 8601 YYYY-MM-DD). Must be after 1900-01-01.
policyholderName	true	string	Name of the policyholder.

DeletePoliciesInput (json)

```
{
  "required": [
    "policyNumber",
    "coverStartDate",
    "policyholderName"
  ],
  "type": "object",
  "properties": {
    "policyNumber": {
      "type": "string",
      "description": "Unique policy reference number. Must not have leading or trailing whitespace."
    },
    "supplementaryPolicyReference": {
      "type": "string",
      "description": "Optional supplementary reference. Must not have leading or trailing whitespace."
    },
    "coverStartDate": {
      "type": "string",
      "description": "Policy cover start date (ISO 8601 YYYY-MM-DD). Must be after 1900-01-01.",
      "format": "date"
    },
    "policyholderName": {
      "type": "string",
      "description": "Name of the policyholder."
    }
  }
}
```

DeletePoliciesResponse

Name	Required	Type	Description
uploadId	true	string (uuid)	Unique identifier for this deletion request
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of when the request was received

DeletePoliciesResponse (json)

```
{
  "required": [
```

```

    "uploadId",
    "timestamp"
  ],
  "type": "object",
  "properties": {
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for this deletion request",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "UTC ISO 8601 timestamp of when the request was
received",
      "format": "date-time"
    }
  }
}

```

ApiResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponse (json)

```

{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",

```

```

        "UNAUTHORISED",
        "FORBIDDEN",
        "NOT_FOUND",
        "INTERNAL_ERROR"
    ],
    "type": "string",
    "description": "Machine-readable error code"
},
"message": {
    "type": "string",
    "description": "Human-readable error message"
},
"details": {
    "description": "Additional error details (e.g. validation
issues)",
    "nullable": true
}
}
}
}
}
}

```

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```

{
    "required": [
        "error"
    ],
    "type": "object",
    "properties": {
        "error": {
            "required": [
                "code",
                "message"
            ],
            "type": "object",
            "properties": {
                "code": {
                    "type": "string",
                    "example": "RATE_LIMIT_EXCEEDED"
                },
                "message": {
                    "type": "string",

```

"example": "Too many requests. Please retry after the period indicated in the Retry-After header."

```
}  
  }  
    }  
      }  
        }
```

Get Test Policy File Upload Status

Returns the current processing status of a previously submitted test policy file upload, including validation results when available. Behaviour is identical to the production endpoint but reads from the isolated test environment.

This endpoint is idempotent.

Recommended polling interval: Poll this endpoint no more than once every 30 seconds to check for status updates. Most uploads complete within a few minutes, but processing time depends on file size and system load. Excessive polling may result in rate limiting.

Endpoint:

GET <https://test-api.eltohub.org.uk/v1/test/insurers/{insurerId}/policies/status/{uploadId}>

Response: 200 OK

Test policy file status returned

ExternalPolicyStatusResponse

Name	Required	Type	Description
id	true	integer	Internal record identifier
uploadId	true	string (uuid)	Unique identifier for the upload request
timestamp	true	string (date-time)	Timestamp when the upload request was created
uploadType	true	string	The method used to upload the policy data
fileName	false	string	Name of the uploaded file
fileType	true	string	The format of the uploaded file
operationType	true	string	The type of operation being performed
status	true	string	Current processing status of the upload
fileRejectionReason	false	string	Reason the file was rejected, if applicable

Name	Required	Type	Description
totalPolicies	false	integer	Total number of policies found in the uploaded file
validPolicies	false	integer	Number of policies that passed validation
rejectedPolicies	false	integer	Number of policies that failed validation
validationErrors	true	ValidationError[]	List of validation errors found during processing
processingStartedAt	false	string (date-time)	Timestamp when processing started
validationCompletedAt	false	string (date-time)	Timestamp when validation completed
processingCompletedAt	false	string (date-time)	Timestamp when processing completed

ExternalPolicyStatusResponse (json)

```
{
  "required": [
    "id",
    "uploadId",
    "timestamp",
    "uploadType",
    "fileType",
    "operationType",
    "status",
    "validationErrors"
  ],
  "type": "object",
  "properties": {
    "id": {
      "type": "integer",
      "description": "Internal record identifier"
    },
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for the upload request",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "Timestamp when the upload request was created",
      "format": "date-time"
    },
    "uploadType": {
      "enum": [
        "WEB",
        "SYNC",
        "API_FILE",
        "API_DIRECT"
      ]
    }
  }
}
```

```

    ],
    "type": "string",
    "description": "The method used to upload the policy data"
  },
  "fileName": {
    "type": "string",
    "description": "Name of the uploaded file",
    "nullable": true
  },
  "fileType": {
    "enum": [
      "JSON",
      "TEXT"
    ],
    "type": "string",
    "description": "The format of the uploaded file"
  },
  "operationType": {
    "enum": [
      "UPSERT",
      "DELETE"
    ],
    "type": "string",
    "description": "The type of operation being performed"
  },
  "status": {
    "enum": [
      "WAITING_FOR_FILE",
      "QUEUED",
      "PROCESSING",
      "COMPLETE",
      "REJECTED",
      "SYSTEM_ERROR"
    ],
    "type": "string",
    "description": "Current processing status of the upload"
  },
  "fileRejectionReason": {
    "type": "string",
    "description": "Reason the file was rejected, if applicable",
    "nullable": true
  },
  "totalPolicies": {
    "type": "integer",
    "description": "Total number of policies found in the uploaded
file",
    "nullable": true
  },
  "validPolicies": {
    "type": "integer",
    "description": "Number of policies that passed validation",
    "nullable": true
  },
  "rejectedPolicies": {
    "type": "integer",
    "description": "Number of policies that failed validation",

```

```

        "nullable": true
    },
    "validationErrors": {
        "type": "array",
        "items": {
            "$ref": "#/components/schemas/ValidationError"
        },
        "description": "List of validation errors found during processing"
    },
    "processingStartedAt": {
        "type": "string",
        "description": "Timestamp when processing started",
        "format": "date-time",
        "nullable": true
    },
    "validationCompletedAt": {
        "type": "string",
        "description": "Timestamp when validation completed",
        "format": "date-time",
        "nullable": true
    },
    "processingCompletedAt": {
        "type": "string",
        "description": "Timestamp when processing completed",
        "format": "date-time",
        "nullable": true
    }
}
}
}

```

default (json)

```

{
    "id": 0,
    "uploadId": "string",
    "timestamp": "string",
    "uploadType": "WEB",
    "fileName": "string",
    "fileType": "JSON",
    "operationType": "UPSERT",
    "status": "WAITING_FOR_FILE",
    "fileRejectionReason": "string",
    "totalPolicies": 0,
    "validPolicies": 0,
    "rejectedPolicies": 0,
    "validationErrors": [{
        "description": "string",
        "path": "string",
        "policyNumber": "string",
        "supplementaryPolicyReference": "string",
        "coverStartDate": "string",
        "policyholderName": "string",
        "employerName": "string",
        "lineNumber": 0
    }],
    "processingStartedAt": "string",
    "validationCompletedAt": "string",
}

```


Response: 400 Bad Request

Invalid uploadId format

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        }
      },
      "details": {
```

```

        "description": "Additional error details (e.g. validation
issues)",
        "nullable": true
    }
}
}
}
}
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "VALIDATION_FAILED",
    "message": "Validation failed",
    "details": {
      "issues": [{
        "path": "",
        "message": "Invalid UUID provided"
      }]
    }
  }
}

```

Response: 401 Unauthorized

Unauthenticated

ApiResponseResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponseResponse (json)

```

{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [

```

```

        "code",
        "message"
    ],
    "type": "object",
    "properties": {
        "code": {
            "enum": [
                "BAD_REQUEST",
                "VALIDATION_FAILED",
                "UNAUTHORISED",
                "FORBIDDEN",
                "NOT_FOUND",
                "INTERNAL_ERROR"
            ],
            "type": "string",
            "description": "Machine-readable error code"
        },
        "message": {
            "type": "string",
            "description": "Human-readable error message"
        },
        "details": {
            "description": "Additional error details (e.g. validation
issues)",
            "nullable": true
        }
    }
}
}
}
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "UNAUTHORISED",
    "message": "Unauthorised"
  }
}

```

Response: 403 Forbidden

Authenticated credentials do not match the requested insurerId

ApiResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code

Name	Required	Type	Description
error.message	true	string	Human-readable error message

ApiResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation issues)",
          "nullable": true
        }
      }
    }
  }
}
```

default (json)

```
{
  "requestId": "abc-123",
  "error": {
    "code": "FORBIDDEN",
    "message": "You do not have permission to access this insurer"
  }
}
```

Response: 404 Not Found

Policy file not found for the given uploadId

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        }
      },
      "details": {
```

```

        "description": "Additional error details (e.g. validation
issues)",
        "nullable": true
    }
}
}
}
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "NOT_FOUND",
    "message": "Policy file not found"
  }
}

```

Response: 500 Internal Server Error

Unexpected internal error

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```

{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {

```

```

        "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
        ],
        "type": "string",
        "description": "Machine-readable error code"
    },
    "message": {
        "type": "string",
        "description": "Human-readable error message"
    },
    "details": {
        "description": "Additional error details (e.g. validation
issues)",
        "nullable": true
    }
}
}
}
}
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "INTERNAL_ERROR",
    "message": "Internal Server Error"
  }
}

```

Response: 429 Too many requests

Rate limit exceeded

Response headers

Name	Required	Type	Description
Retry-After	false	integer	Number of seconds to wait before retrying
X-RateLimit-Remaining-Minute	false	integer	Remaining requests allowed in the current minute window

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```

default (json)

```
{
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Too many requests. Please retry after the period indicated
in the Retry-After header."
  }
}
```

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}
```

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```

ExternalPolicyStatusResponse

Name	Required	Type	Description
id	true	integer	Internal record identifier
uploadId	true	string (uuid)	Unique identifier for the upload request

Name	Required	Type	Description
timestamp	true	string (date-time)	Timestamp when the upload request was created
uploadType	true	string	The method used to upload the policy data
fileName	false	string	Name of the uploaded file
fileType	true	string	The format of the uploaded file
operationType	true	string	The type of operation being performed
status	true	string	Current processing status of the upload
fileRejectionReason	false	string	Reason the file was rejected, if applicable
totalPolicies	false	integer	Total number of policies found in the uploaded file
validPolicies	false	integer	Number of policies that passed validation
rejectedPolicies	false	integer	Number of policies that failed validation
validationErrors	true	ValidationError[]	List of validation errors found during processing
processingStartedAt	false	string (date-time)	Timestamp when processing started
validationCompletedAt	false	string (date-time)	Timestamp when validation completed
processingCompletedAt	false	string (date-time)	Timestamp when processing completed

ExternalPolicyStatusResponse (json)

```
{
  "required": [
    "id",
    "uploadId",
    "timestamp",
    "uploadType",
    "fileType",
    "operationType",
    "status",
    "validationErrors"
  ],
  "type": "object",
  "properties": {
```

```
"id": {
  "type": "integer",
  "description": "Internal record identifier"
},
"uploadId": {
  "type": "string",
  "description": "Unique identifier for the upload request",
  "format": "uuid"
},
"timestamp": {
  "type": "string",
  "description": "Timestamp when the upload request was created",
  "format": "date-time"
},
"uploadType": {
  "enum": [
    "WEB",
    "SYNC",
    "API_FILE",
    "API_DIRECT"
  ],
  "type": "string",
  "description": "The method used to upload the policy data"
},
"fileName": {
  "type": "string",
  "description": "Name of the uploaded file",
  "nullable": true
},
"fileType": {
  "enum": [
    "JSON",
    "TEXT"
  ],
  "type": "string",
  "description": "The format of the uploaded file"
},
"operationType": {
  "enum": [
    "UPSERT",
    "DELETE"
  ],
  "type": "string",
  "description": "The type of operation being performed"
},
"status": {
  "enum": [
    "WAITING_FOR_FILE",
    "QUEUED",
    "PROCESSING",
    "COMPLETE",
    "REJECTED",
    "SYSTEM_ERROR"
  ],
  "type": "string",
  "description": "Current processing status of the upload"
}
```

```

    },
    "fileRejectionReason": {
      "type": "string",
      "description": "Reason the file was rejected, if applicable",
      "nullable": true
    },
    "totalPolicies": {
      "type": "integer",
      "description": "Total number of policies found in the uploaded
file",
      "nullable": true
    },
    "validPolicies": {
      "type": "integer",
      "description": "Number of policies that passed validation",
      "nullable": true
    },
    "rejectedPolicies": {
      "type": "integer",
      "description": "Number of policies that failed validation",
      "nullable": true
    },
    "validationErrors": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "description": "List of validation errors found during processing"
    },
    "processingStartedAt": {
      "type": "string",
      "description": "Timestamp when processing started",
      "format": "date-time",
      "nullable": true
    },
    "validationCompletedAt": {
      "type": "string",
      "description": "Timestamp when validation completed",
      "format": "date-time",
      "nullable": true
    },
    "processingCompletedAt": {
      "type": "string",
      "description": "Timestamp when processing completed",
      "format": "date-time",
      "nullable": true
    }
  }
}

```

ValidationError

Name	Required	Type	Description
description	true	string	Description of the validation error
path	true	string	Path to the field that caused the error
policyNumber	false	string	Policy number associated with the error
supplementaryPolicyReference	false	string	Supplementary policy reference associated with the error
coverStartDate	false	string (date-time)	Cover start date associated with the error
policyholderName	false	string	Policyholder name associated with the error
employerName	false	string	Employer name associated with the error
lineNumber	false	integer	Line number in the file where the error occurred

ValidationError (json)

```
{
  "required": [
    "description",
    "path"
  ],
  "type": "object",
  "properties": {
    "description": {
      "type": "string",
      "description": "Description of the validation error"
    },
    "path": {
      "type": "string",
      "description": "Path to the field that caused the error"
    },
    "policyNumber": {
      "type": "string",
      "description": "Policy number associated with the error",
      "nullable": true
    },
    "supplementaryPolicyReference": {
      "type": "string",
      "description": "Supplementary policy reference associated with the error",

```

```
    "nullable": true
  },
  "coverStartDate": {
    "type": "string",
    "description": "Cover start date associated with the error",
    "format": "date-time",
    "nullable": true
  },
  "policyholderName": {
    "type": "string",
    "description": "Policyholder name associated with the error",
    "nullable": true
  },
  "employerName": {
    "type": "string",
    "description": "Employer name associated with the error",
    "nullable": true
  },
  "lineNumber": {
    "type": "integer",
    "description": "Line number in the file where the error occurred",
    "nullable": true
  }
}
```

Health Check

Endpoint:

GET <https://test-api.eltohub.org.uk/v1/test/health>

This endpoint is idempotent.

Response: 200 OK

Service is healthy

HealthResponse

Name	Required	Type	Description
apiStatus	true	string	Overall API health status
dbStatus	true	string	Database connectivity status
adlsStatus	true	string	ADLS storage connectivity status
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of the health check
version	true	string	API version
uptime	true	integer	Seconds since service start
error	false	string	Error message when an unexpected failure occurs during the health check

HealthResponse (json)

```
{
  "required": [
    "apiStatus",
    "dbStatus",
    "adlsStatus",
    "timestamp",
    "version",
    "uptime"
  ],
  "type": "object",
  "properties": {
    "apiStatus": {
      "enum": [
        "healthy",
        "unhealthy"
      ],
      "type": "string",
      "description": "Overall API health status"
    }
  }
}
```



```

    },
    "dbStatus": {
      "enum": [
        "healthy",
        "unhealthy",
        "unknown"
      ],
      "type": "string",
      "description": "Database connectivity status"
    },
    "adlsStatus": {
      "enum": [
        "healthy",
        "unhealthy",
        "unknown"
      ],
      "type": "string",
      "description": "ADLS storage connectivity status"
    },
    "timestamp": {
      "type": "string",
      "description": "UTC ISO 8601 timestamp of the health check",
      "format": "date-time"
    },
    "version": {
      "type": "string",
      "description": "API version"
    },
    "uptime": {
      "type": "integer",
      "description": "Seconds since service start"
    },
    "error": {
      "type": "string",
      "description": "Error message when an unexpected failure occurs during the health check"
    }
  }
}

```

default (json)

```

{
  "apiStatus": "healthy",
  "dbStatus": "healthy",
  "adlsStatus": "healthy",
  "timestamp": "string",
  "version": "string",
  "uptime": 0,
  "error": "string"
}

```

Response: 503 Service Unavailable

Service is unhealthy

HealthResponse

Name	Required	Type	Description
apiStatus	true	string	Overall API health status
dbStatus	true	string	Database connectivity status
adlsStatus	true	string	ADLS storage connectivity status
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of the health check
version	true	string	API version
uptime	true	integer	Seconds since service start
error	false	string	Error message when an unexpected failure occurs during the health check

HealthResponse (json)

```
{
  "required": [
    "apiStatus",
    "dbStatus",
    "adlsStatus",
    "timestamp",
    "version",
    "uptime"
  ],
  "type": "object",
  "properties": {
    "apiStatus": {
      "enum": [
        "healthy",
        "unhealthy"
      ],
      "type": "string",
      "description": "Overall API health status"
    },
    "dbStatus": {
      "enum": [
        "healthy",
        "unhealthy",
        "unknown"
      ],
      "type": "string",
      "description": "Database connectivity status"
    },
    "adlsStatus": {
      "enum": [
        "healthy",

```

```

        "unhealthy",
        "unknown"
    ],
    "type": "string",
    "description": "ADLS storage connectivity status"
},
"timestamp": {
    "type": "string",
    "description": "UTC ISO 8601 timestamp of the health check",
    "format": "date-time"
},
"version": {
    "type": "string",
    "description": "API version"
},
"uptime": {
    "type": "integer",
    "description": "Seconds since service start"
},
"error": {
    "type": "string",
    "description": "Error message when an unexpected failure occurs
during the health check"
}
}
}

```

default (json)

```

{
    "apiStatus": "healthy",
    "dbStatus": "healthy",
    "adlsStatus": "healthy",
    "timestamp": "string",
    "version": "string",
    "uptime": 0,
    "error": "string"
}

```

Response: 429 Too many requests

Rate limit exceeded

Response headers

Name	Required	Type	Description
Retry-After	false	integer	Number of seconds to wait before retrying
X-RateLimit-Remaining-Minute	false	integer	Remaining requests allowed in the current minute window

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```

default (json)

```
{
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Too many requests. Please retry after the period indicated
in the Retry-After header."
  }
}
```

HealthResponse

Name	Required	Type	Description
apiStatus	true	string	Overall API health status

Name	Required	Type	Description
dbStatus	true	string	Database connectivity status
adlsStatus	true	string	ADLS storage connectivity status
timestamp	true	string (date-time)	UTC ISO 8601 timestamp of the health check
version	true	string	API version
uptime	true	integer	Seconds since service start
error	false	string	Error message when an unexpected failure occurs during the health check

HealthResponse (json)

```
{
  "required": [
    "apiStatus",
    "dbStatus",
    "adlsStatus",
    "timestamp",
    "version",
    "uptime"
  ],
  "type": "object",
  "properties": {
    "apiStatus": {
      "enum": [
        "healthy",
        "unhealthy"
      ],
      "type": "string",
      "description": "Overall API health status"
    },
    "dbStatus": {
      "enum": [
        "healthy",
        "unhealthy",
        "unknown"
      ],
      "type": "string",
      "description": "Database connectivity status"
    },
    "adlsStatus": {
      "enum": [
        "healthy",
        "unhealthy",
        "unknown"
      ],
      "type": "string",
      "description": "ADLS storage connectivity status"
    }
  }
}
```

```

    },
    "timestamp": {
      "type": "string",
      "description": "UTC ISO 8601 timestamp of the health check",
      "format": "date-time"
    },
    "version": {
      "type": "string",
      "description": "API version"
    },
    "uptime": {
      "type": "integer",
      "description": "Seconds since service start"
    },
    "error": {
      "type": "string",
      "description": "Error message when an unexpected failure occurs
during the health check"
    }
  }
}

```

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```

{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}

```



Request a Test Policy File Upload

Submits metadata for a test policy file upload and returns a SAS upload URL. Behaviour is identical to the production endpoint but writes to an isolated test environment — no live data is affected.

This endpoint is not idempotent.

How to upload your file

1. Call this endpoint with your file metadata (fileName, fileType, operationType).
2. On success, you will receive an uploadUrl (a pre-signed SAS URL).
3. Upload your file to that URL using an HTTP **PUT** request with the following required header:
 - o x-ms-blob-type: BlockBlob
4. The Content-Type header should match your file format (e.g. application/json for JSON files, text/plain for TEXT files).
5. The upload URL is **write-only** and expires after 60 minutes (see expiresAt in the response).
6. Once your file is uploaded, it will be validated and processed asynchronously.

Example using cURL

```
curl -X PUT \  
-H "x-ms-blob-type: BlockBlob" \  
-H "Content-Type: application/json" \  
--data-binary @yourfile.json \  
"<uploadUrl from response>"
```

Example using PowerShell

```
Invoke-RestMethod -Method Put -Uri $uploadUrl -InFile .\yourfile.json -Headers @{ 'x-ms-blob-type' = 'BlockBlob'; 'Content-Type' = 'application/json' }
```

A successful upload produces no output — this is expected and means your file was received. If the upload fails, you will receive an error message. If the URL has expired, simply request a new upload URL and try again.

Endpoint:

POST <https://test-api.eltohub.org.uk/v1/test/insurers/{insurerId}/policies/file>

Try this operation

Request

Request parameters

Name	In	Required	Type	Description
insurerId	template	true	string	The insurer identifier

Request body

PolicyUploadRequest

Name	Required	Type	Description
fileName	true	string	Name of the file (letters and numbers only, no extension)
fileType	true	string	The format of the file being uploaded
operationType	true	string	The type of operation to perform with the uploaded data

PolicyUploadRequest (json)

```
{
  "required": [
    "fileName",
    "fileType",
    "operationType"
  ],
  "type": "object",
  "properties": {
    "fileName": {
      "pattern": "^[a-zA-Z0-9]+$",
      "type": "string",
      "description": "Name of the file (letters and numbers only, no
extension)"
    },
    "fileType": {
      "enum": [
        "JSON",
        "TEXT"
      ],
      "type": "string",
      "description": "The format of the file being uploaded"
    },
    "operationType": {
      "enum": [
        "UPSERT"
      ],
      "type": "string",
      "description": "The type of operation to perform with the uploaded
data"
    }
  }
}
```

default (json)

```
{
  "fileName": "string",
  "fileType": "JSON",
  "operationType": "UPSERT"
}
```

Response: 200 OK

Test upload request accepted; SAS upload URL returned

PolicyUploadResponse

Name	Required	Type	Description
uploadId	true	string (uuid)	Unique identifier for this upload request
timestamp	true	string (date-time)	Timestamp when the upload request was created
expiresAt	true	string (date-time)	Expiry time of the SAS upload URL
uploadUrl	true	string (uri)	Pre-signed SAS URL to upload the file to. Send an HTTP PUT request to this URL with your file as the...

PolicyUploadResponse (json)

```
{
  "required": [
    "uploadId",
    "timestamp",
    "expiresAt",
    "uploadUrl"
  ],
  "type": "object",
  "properties": {
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for this upload request",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "Timestamp when the upload request was created",
      "format": "date-time"
    },
    "expiresAt": {
      "type": "string",
      "description": "Expiry time of the SAS upload URL",
      "format": "date-time"
    },
    "uploadUrl": {
```

```

        "type": "string",
        "description": "Pre-signed SAS URL to upload the file to. Send an
HTTP PUT request to this URL with your file as the body and the header `x-ms-
blob-type: BlockBlob`. The URL is write-only and expires at the time indicated
by `expiresAt`. A successful upload returns HTTP 201.",
        "format": "uri"
    }
}
}

```

default (json)

```

{
  "uploadId": "string",
  "timestamp": "string",
  "expiresAt": "string",
  "uploadUrl": "string"
}

```

Response: 400 Bad Request

Validation failed or malformed request body

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```

{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {

```

```

        "code": {
            "enum": [
                "BAD_REQUEST",
                "VALIDATION_FAILED",
                "UNAUTHORISED",
                "FORBIDDEN",
                "NOT_FOUND",
                "INTERNAL_ERROR"
            ],
            "type": "string",
            "description": "Machine-readable error code"
        },
        "message": {
            "type": "string",
            "description": "Human-readable error message"
        },
        "details": {
            "description": "Additional error details (e.g. validation
issues)",
            "nullable": true
        }
    }
}

```

badRequest (json)

```

{
    "requestId": "abc-123",
    "error": {
        "code": "BAD_REQUEST",
        "message": "Request body is missing or invalid"
    }
}

```

validationFailed (json)

```

{
    "requestId": "abc-123",
    "error": {
        "code": "VALIDATION_FAILED",
        "message": "Validation failed",
        "details": {
            "issues": [{
                "path": "fileName",
                "message": "fileName is required"
            }]
        }
    }
}

```

Response: 401 Unauthorized

Unauthenticated

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}
```

```
}
}
}
}
```

default (json)

```
{
  "requestId": "abc-123",
  "error": {
    "code": "UNAUTHORISED",
    "message": "Unauthorised"
  }
}
```

Response: 403 Forbidden

Authenticated credentials do not match the requested insurerId

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",

```

```

        "UNAUTHORISED",
        "FORBIDDEN",
        "NOT_FOUND",
        "INTERNAL_ERROR"
    ],
    "type": "string",
    "description": "Machine-readable error code"
},
"message": {
    "type": "string",
    "description": "Human-readable error message"
},
"details": {
    "description": "Additional error details (e.g. validation
issues)",
    "nullable": true
}
}
}
}
}
}

```

default (json)

```

{
    "requestId": "abc-123",
    "error": {
        "code": "FORBIDDEN",
        "message": "You do not have permission to access this insurer"
    }
}

```

Response: 500 Internal Server Error

Unexpected internal error

ApiErrorResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiErrorResponse (json)

```

{
    "required": [
        "requestId",
        "error"
    ],
    "type": "object",
    "properties": {

```

```

    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}

```

default (json)

```

{
  "requestId": "abc-123",
  "error": {
    "code": "INTERNAL_ERROR",
    "message": "Internal Server Error"
  }
}

```

Response: 429 Too many requests

Rate limit exceeded

Response headers

Name	Required	Type	Description
Retry-After	false	integer	Number of seconds to wait before retrying
X-RateLimit-Remaining-Minute	false	integer	Remaining requests allowed in the current minute window

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```

default (json)

```
{
  "error": {
    "code": "RATE_LIMIT_EXCEEDED",
    "message": "Too many requests. Please retry after the period indicated
in the Retry-After header."
  }
}
```

PolicyUploadRequest

Name	Required	Type	Description
fileName	true	string	Name of the file (letters and numbers only, no extension)
fileType	true	string	The format of the file being uploaded
operationType	true	string	The type of operation to perform with the uploaded data

PolicyUploadRequest (json)

```
{
  "required": [
    "fileName",
    "fileType",
    "operationType"
  ],
  "type": "object",
  "properties": {
    "fileName": {
      "pattern": "^[a-zA-Z0-9]+$",
      "type": "string",
      "description": "Name of the file (letters and numbers only, no
extension)"
    },
    "fileType": {
      "enum": [
        "JSON",
        "TEXT"
      ],
      "type": "string",
      "description": "The format of the file being uploaded"
    },
    "operationType": {
      "enum": [
        "UPSERT"
      ],
      "type": "string",
      "description": "The type of operation to perform with the uploaded
data"
    }
  }
}
```

PolicyUploadResponse

Name	Required	Type	Description
uploadId	true	string (uuid)	Unique identifier for this upload request
timestamp	true	string (date-time)	Timestamp when the upload request was created
expiresAt	true	string (date-time)	Expiry time of the SAS upload URL
uploadUrl	true	string (uri)	Pre-signed SAS URL to upload the file to. Send an HTTP PUT request to this URL with your file as the body and the header x-ms-blob-type: BlockBlob. The URL is write-only and expires at the time indicated by expiresAt. A successful upload returns HTTP 201.

PolicyUploadResponse (json)

```
{
  "required": [
    "uploadId",
    "timestamp",
    "expiresAt",
    "uploadUrl"
  ],
  "type": "object",
  "properties": {
    "uploadId": {
      "type": "string",
      "description": "Unique identifier for this upload request",
      "format": "uuid"
    },
    "timestamp": {
      "type": "string",
      "description": "Timestamp when the upload request was created",
      "format": "date-time"
    },
    "expiresAt": {
      "type": "string",
      "description": "Expiry time of the SAS upload URL",
      "format": "date-time"
    },
    "uploadUrl": {
      "type": "string",
      "description": "Pre-signed SAS URL to upload the file to. Send an HTTP PUT request to this URL with your file as the body and the header `x-ms-blob-type: BlockBlob`. The URL is write-only and expires at the time indicated by `expiresAt`. A successful upload returns HTTP 201.",
      "format": "uri"
    }
  }
}
```

ApiResponse

Name	Required	Type	Description
requestId	true	string	Unique identifier for the request
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

ApiResponse (json)

```
{
  "required": [
    "requestId",
    "error"
  ],
  "type": "object",
  "properties": {
    "requestId": {
      "type": "string",
      "description": "Unique identifier for the request"
    },
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "enum": [
            "BAD_REQUEST",
            "VALIDATION_FAILED",
            "UNAUTHORISED",
            "FORBIDDEN",
            "NOT_FOUND",
            "INTERNAL_ERROR"
          ],
          "type": "string",
          "description": "Machine-readable error code"
        },
        "message": {
          "type": "string",
          "description": "Human-readable error message"
        },
        "details": {
          "description": "Additional error details (e.g. validation
issues)",
          "nullable": true
        }
      }
    }
  }
}
```

RateLimitErrorResponse

Name	Required	Type	Description
error.code	true	string	Machine-readable error code
error.message	true	string	Human-readable error message

RateLimitErrorResponse (json)

```
{
  "required": [
    "error"
  ],
  "type": "object",
  "properties": {
    "error": {
      "required": [
        "code",
        "message"
      ],
      "type": "object",
      "properties": {
        "code": {
          "type": "string",
          "example": "RATE_LIMIT_EXCEEDED"
        },
        "message": {
          "type": "string",
          "example": "Too many requests. Please retry after the
period indicated in the Retry-After header."
        }
      }
    }
  }
}
```