

# Ingestion Service – API Contracts

**Base URL:** `http://localhost:8082/api/v1/ingestion`

**Service Port:** 8082

**Document Version:** 1.0

## Table of Contents

- Phase 3: File Upload & Job Creation
- Phase 4: Processing Pipeline Integration
- Phase 7: UI Polling (Upload Status)
- Error Response Contract
- Status Enum & State Machine

## Phase 3: File Upload & Job Creation

User → BFF → Ingestion. Accepts file, validates, stores, creates job in MongoDB.

POST `/api/v1/ingestion/upload`

**POST** `/api/v1/ingestion/upload`

**Content-Type:** `multipart/form-data`

Parameter	Type	Required	Description
file	File	Yes	.xls, .xlsx, or .csv; max 50MB
insurerId	String	Yes	Insurer identifier
uploadedBy	String	Yes	User or system performing upload

**Success Response:** `201 Created`

```
{
  "jobId": "uuid-string",
  "status": "UPLOADED"
}
```

Field	Type	Description
jobId	String (UUID)	Unique job identifier
status	Enum	UPLOADED

### Error Responses:

Code	Condition
400	Invalid file type, missing filename, empty file, file > 50MB
500	Storage failure

## Phase 4 & Phase 7: Get Job Status

**Phase 4:** Processing Service fetches filePath, insurerId to process the file.

**Phase 7:** BFF proxies to Ingestion for UI progress polling.

GET /api/v1/ingestion/status/{jobId}

GET /api/v1/ingestion/status/{jobId}

Path Parameter	Type	Description
jobId	String (UUID)	Ingestion job identifier

**Success Response:** 200 OK

```
{
  "jobId": "uuid-string",
  "status": "UPLOADED",
  "processedRecords": 0,
  "totalRecords": 0,
}
```

```
"filePath": "C:\\...\\storage\\ingestion\\{jobId}.csv",
"insurerId": "HDFC_LIFE",
"createdAt": "2026-02-12T17:02:42.087",
"updatedAt": "2026-02-12T17:02:42.087"
}
```

Field	Type	Description
jobId	String	Job identifier
status	Enum	UPLOADED, PROCESSING, COMPLETED, FAILED
processedRecords	int	Records processed so far
totalRecords	int	Total records in file
filePath	String	Absolute path to stored file
insurerId	String	Insurer identifier
createdAt	DateTime (ISO-8601)	Job creation timestamp
updatedAt	DateTime (ISO-8601)	Last update timestamp

#### Error Responses:

Code	Condition
400	Job not found
500	Internal error

## Phase 4: Processing Pipeline – Status & Progress Updates

Processing Service transitions job state and updates progress.

PATCH /api/v1/ingestion/{jobId}/status

**PATCH** /api/v1/ingestion/{jobId}/status

**Content-Type:** application/json

Path Parameter	Type	Description
jobId	String (UUID)	Ingestion job identifier

**Request Body:**

```
{
  "status": "PROCESSING",
  "failureReason": null
}
```

Field	Type	Required	Description
status	Enum	Yes	PROCESSING, COMPLETED, or FAILED
failureReason	String	No	Required when status=FAILED

**Success Response:** 204 No Content

**Allowed Transitions:**

From	To
UPLOADED	PROCESSING
PROCESSING	COMPLETED or FAILED
COMPLETED / FAILED	None (terminal)

**Error Responses:**

Code	Condition
400	Job not found, status missing
409	Invalid state transition
500	Internal error

PATCH /api/v1/ingestion/{jobId}/progress

**PATCH** /api/v1/ingestion/{jobId}/progress

**Content-Type:** application/json

Path Parameter	Type	Description
jobId	String (UUID)	Ingestion job identifier

**Request Body:**

```
{
  "processedRecordsDelta": 10
}
```

Field	Type	Required	Constraints	Description
processedRecordsDelta	int	Yes	> 0	Increment to add to processedRecords

**Success Response:** 204 No Content

**Error Responses:**

Code	Condition
400	Job not found, processedRecordsDelta ≤ 0
409	Job not in PROCESSING state
500	Internal error

*Note: Idempotent when retried with same delta.*

**Error Response Contract**

All error responses use this structure.

**Error Response Body**

```
{
  "error": "Error message string"
}
```

HTTP Code	Exception	Use Case
-----------	-----------	----------

400 Bad Request	IllegalArgumentException	Validation failures
409 Conflict	IllegalStateException	Invalid state transitions
500 Internal Server Error	RuntimeException	Unexpected failures

## Status Enum & State Machine

### IngestionStatus Enum

Value	Description
UPLOADED	File stored, job created
PROCESSING	Records being processed
COMPLETED	Processing finished successfully
FAILED	Processing failed

### State Machine

UPLOADED → PROCESSING → COMPLETED  
                                  └→ FAILED

No backward transitions. No skipping PROCESSING. COMPLETED and FAILED are terminal.

## API Summary by Phase

Phase	API	Method	Purpose
3	/upload	POST	Accept file, create job
4	/status/{jobId}	GET	Fetch filePath, insurerId for processing

4	/jobId/status	PATCH	Transition job state
4	/jobId/progress	PATCH	Increment processedRecords
7	/status/{jobId}	GET	Progress for UI polling (via BFF)