

Data Pipeline Service – API Contracts

Document Version: 1.0

Last Updated: February 2026

Base URL: `http://localhost:8082`

This document details all API contracts: endpoints exposed by the data-pipeline-service and those it consumes from external services.

1. APIs Exposed by Data Pipeline Service

1.1 Public Ingestion API (`/api/public/v1/ingestion`)

No authentication required. Used by Insurer Portal and external integrators.

POST `/api/public/v1/ingestion/upload`

Upload a policy file (CSV or Excel) for ingestion.

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

Parameter	Type	Required	Description
file	File	Yes	CSV or Excel (.csv, .xls, .xlsx), max 50MB
insurerId	String	Yes	One of: HEALTH_INSURER, AUTO_INSURER, LIFE_INSURER
uploadedBy	String	Yes	Identifier of the uploader (e.g., email)
fileType	String	No	normal (default) or correction

Response: 201 Created

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

Error responses: - 400 – Validation failed (invalid file, missing columns, schema mismatch) - 500 – Storage or unexpected error

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

Get the status and details of an ingestion job.

Path Parameters: | Parameter | Type | Description | |-----|-----|-----| | jobId | String | Ingestion job ID |

Response: 200 OK

```
{ "jobId": "uuid-string", "status": "COMPLETED", "processedRecords": 95,
  "totalRecords": 100, "filePath": "/abs/path/storage/ingestion/{jobId}.csv",
  "insurerId": "HEALTH_INSURER", "fileType": "normal", "createdAt":
  "2026-02-26T10:00:00", "updatedAt": "2026-02-26T10:05:00", "failureReason":
  null, "verificationFailures": [ { "policyNumber": "HEPOL100001", "reason": "No
  customer found (mobile/email/PAN)" } ] }
```

Status values: UPLOADED, PROCESSING, COMPLETED, FAILED

Error responses: - 400 – Job not found

GET /api/public/v1/ingestion/jobs

List all ingestion jobs from MongoDB.

Response: 200 OK

```
[ { "jobId": "uuid-1", "insurerId": "HEALTH_INSURER", "status": "COMPLETED",
  "totalRecords": 100, "processedRecords": 95, "createdAt": "2026-02-26T10:00:00"
  } ]
```

POST /api/public/v1/ingestion/process/{jobId}

Trigger processing of an uploaded file.

Path Parameters: | Parameter | Type | Description | |-----|-----|-----| | jobId | String | Ingestion job ID |

Query Parameters: | Parameter | Type | Required | Description |
|-----|-----|-----|-----| | policyType | String | No | e.g.
HEALTH, MOTOR, TERM_LIFE. Resolved from config if omitted |

Response: 202 Accepted

```
{ "jobId": "uuid-string", "message": "Processing started", "status":
  "PROCESSING" }
```

Error responses: - 400 – Job not in UPLOADED state - 500 – Processing failed

1.2 Authenticated Ingestion API (/api/v1/ingestion)

JWT required in `Authorization` header. Uses `ApiResponse<T>` wrapper.

POST `/api/v1/ingestion/upload`

Upload file (JWT-based; `uploadedBy` extracted from token).

Request: - **Headers:** `Authorization: Bearer <jwt>` - **Content-Type:** `multipart/form-data` -

Parameters: Same as public upload (`file`, `insurerId`, `fileType`)

Response: 201 Created

```
{ "success": true, "message": "File uploaded successfully", "data": { "jobId": "uuid-string", "status": "UPLOADED" }, "timestamp": "2026-02-26T10:00:00" }
```

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

Get job status (wrapped in `ApiResponse`).

Response: 200 OK

```
{ "success": true, "message": "Job status retrieved successfully", "data": { "jobId": "uuid-string", "status": "COMPLETED", "processedRecords": 95, "totalRecords": 100, "verificationFailures": [...] } }
```

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

Internal: increment processed records count.

Request body:

```
{ "processedRecordsDelta": 5 }
```

Response: 200 OK

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

Internal: transition job state.

Request body:

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

Allowed transitions: `UPLOADED` → `PROCESSING`; `PROCESSING` → `COMPLETED` | `FAILED`

GET `/api/v1/ingestion/health`

Health check.

Response: 200 OK

```
{ "success": true, "data": "healthy" }
```

1.3 Portfolio API (/api/public/v1/portfolio)

GET /api/public/v1/portfolio

List all customer portfolios (customer + policies).

Response: 200 OK

```
[ { "id": "mongo-id", "customerId": "uuid-string", "firstName": "John",
  "lastName": "Doe", "email": "john@example.com", "mobileNumber": "9876543210",
  "panNumber": "ABCDE1234F", "policies": [ { "policyId": "uuid", "policyNumber":
  "HEPOL100001", "insurerId": "HEALTH_INSURER", "policyType": "HEALTH",
  "planName": "Family Health", "premiumAmount": 5000, "sumAssured": 500000,
  "startDate": "2025-01-01", "endDate": "2026-01-01", "status": "ACTIVE" } ] } ]
```

GET /api/public/v1/portfolio/{customerId}

Get portfolio for a specific customer.

Response: 200 OK Or 404 Not Found

1.4 Metadata API (/api/public/v1/metadata)

GET /api/public/v1/metadata/insurers

List configured insurer IDs.

Response: 200 OK

```
[ "HEALTH_INSURER", "AUTO_INSURER", "LIFE_INSURER" ]
```

GET /api/public/v1/metadata/mappings

Get field mappings for an insurer and policy type.

Query Parameters: | Parameter | Type | Required | Description |

|-----|-----|-----|-----| | insurerId | String | Yes | e.g.

HEALTH_INSURER | | policyType | String | No | e.g. HEALTH. Resolved from config if omitted |

Response: 200 OK

```
{ "insurerId": "HEALTH_INSURER", "policyType": "HEALTH", "fieldMappings": [ {  
  "sourceField": "Policy Number", "targetField": "policyNumber", "dataType":  
  "STRING", "transformFunction": null }, { "sourceField": "Mobile",  
  "targetField": "mobileNumber", "transformFunction": "normalizeMobile" } ],  
  "description": "Source CSV/Excel column names map to canonical fields..." }
```

2. APIs Consumed by Data Pipeline Service

2.1 Customer Service (Port 8081)

Base URL: `http://localhost:8081` (configurable via `customer.service.url`)

GET `/api/v1/customers/{customerId}`

Get customer by ID.

Response: 200 OK

```
{ "customerId": "uuid-string", "firstName": "John", "lastName": "Doe", "email":  
  "john@example.com", "mobileNumber": "9876543210", "panNumber": "ABCDE1234F",  
  "dateOfBirth": "1990-05-15" }
```

GET `/api/v1/customers/search/mobile/{mobile}`

Search customer by mobile number.

Response: 200 OK (customer) or 404 Not Found

GET `/api/v1/customers/search/email/{email}`

Search customer by email.

Response: 200 OK (customer) or 404 Not Found

GET `/api/v1/customers/search/pan/{pan}`

Search customer by PAN number.

Response: 200 OK (customer) or 404 Not Found

2.2 Policy Service (Port 8085)

Base URL: `http://localhost:8085` (configurable via `policy.service.url`)

POST `/api/v1/policies`

Create a policy.

Request body:

```
{ "customerId": "uuid-string", "insurerId": "HEALTH_INSURER", "policyNumber":
"HEPOL100001", "policyType": "HEALTH", "planName": "Family Health",
"premiumAmount": 5000.00, "sumAssured": 500000.00, "startDate": "2025-01-01",
"endDate": "2026-01-01", "status": "ACTIVE" }
```

Required fields: `customerId`, `insurerId`, `policyNumber`, `policyType`, `premiumAmount`, `sumAssured`, `status`

Response: 200 OK

```
{ "id": "uuid-string", "customerId": "uuid-string", "insurerId":
"HEALTH_INSURER", "policyNumber": "HEPOL100001", "policyType": "HEALTH",
"planName": "Family Health", "premiumAmount": 5000.00, "sumAssured": 500000.00,
"startDate": "2025-01-01", "endDate": "2026-01-01", "status": "ACTIVE",
"createdAt": "2026-02-26T10:05:00", "updatedAt": "2026-02-26T10:05:00" }
```

3. Data Types Reference

3.1 IngestionStatus (enum)

Value	Description
UPLOADED	File uploaded, ready for processing
PROCESSING	Processing in progress
COMPLETED	Processing finished successfully
FAILED	Processing failed

3.2 Verification Failure

Field	Type	Description
policyNumber	String	Policy identifier from file
reason	String	Failure reason

3.3 ApiResponse (standard wrapper)

Field	Type	Description
-------	------	-------------

success	boolean	Request success
message	String	Human-readable message
data	T	Response payload
error	Object	Present when success=false
timestamp	String	ISO 8601 timestamp

ErrorDetails: | Field | Type | Description | |-----|-----|-----| | code | String | Error code | | message | String | Error message | | details | String | Optional details |

4. Summary Table

API	Method	Path	Auth
Upload	POST	/api/public/v1/ingestion/upload	None
Job Status	GET	/api/public/v1/ingestion/status/{id}	None
List Jobs	GET	/api/public/v1/ingestion/jobs	None
Process	POST	/api/public/v1/ingestion/process/{id}	None
Portfolio	GET	/api/public/v1/portfolio	None
Portfolio	GET	/api/public/v1/portfolio/customerId	None
Insurers	GET	/api/public/v1/metadata/insurers	None
Mappings	GET	/api/public/v1/metadata/mappings	None
Upload (JWT)	POST	/api/v1/ingestion/upload	JWT
Status (JWT)	GET	/api/v1/ingestion/status/{id}	JWT
Progress	PATCH	/api/v1/ingestion/{id}	JWT
Status	PATCH	/api/v1/ingestion/{id}	JWT
Health	GET	/api/v1/ingestion/health	JWT

End of Document