PhenOM Inference API User Guide

This user guide summarizes how to use the PhenOM Inference API based on the latest spec.

Base URL & Reference

- **API base URL**
  `https://phenom-api.om1.com`

Authentication

- **Auth method**: OAuth2 Bearer token (Auth0)
- **Header format**:

```
Authorization: Bearer <jwt>
```

Your JWT determines which tenant and resources you can access.

Inference Options

You can run predictions in two ways:

- **Single-patient inference**
  - **Best for**: point-of-care workflows and interactive apps
  - **Pattern**: submit a single `patient_history` JSON → get a `job_id` → poll job for results
  - **Endpoint**: `/v1/job/from-patient`
  - **Multiple patients**: use the **batch** workflow (`/v1/batch`) instead of a multi-patient real-time endpoint

- **Batch inference**
  - **Best for**: large populations, scheduled risk assessments, research & analytics
  - **Pattern**: upload files → finalize → (optionally) validate → start job → poll → fetch results
  - **Endpoints**: `/v1/batch`, `/v1/batch/{batch_id}/upload`, `/v1/batch/{batch_id}/finalize`, `/v1/batch/{batch_id}/start`, plus job endpoints under `/v1/job/...`

Quick Start – Single Patient

Use this for a single patient at a time.

Request: create job from patient

```
curl -X POST https://phenom-api.om1.com/v1/job/from-patient \
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '{
        "outcome_ids": [
            "v0p3_65plus_fixed_anchor_2023_JanToDec_04012025_any_time_hospitalization_future_1mon
        ],
        "patient_history": {
            "patient": {
            "patient_id": "abc123",
            "birth_date": "1964-05-18",
            "sex": "F",
            "zip3": "021"
            },
            "lab_results": [
            {
                "lab_date": "2025-07-03",
                "loinc": "718-7",
                "value_abnormal": "N"
            }
            ],
            "diagnoses": [
            {
                "diagnosis_date": "2020-01-01",
                "code": "I10",
                "code_type_name": "ICD10"
            }
            ],
            "procedures": [
            {
                "procedure_date": "2024-12-15",
                "code": "93000",
                "code_type_name": "CPT"
            }
            ],
            "medications": [
            {
                "medication_start": "2023-03-01",
                "code": "C09AA02",
                "code_type": "ATC"
            }
            ]
        },
        "idempotency_key": "idem-b-20251107-abc123"
    }'
```

Immediate response (example):

```
{
  "job_id": "job_01J90Q2TTM",
  "status": "QUEUED"
}
```

Poll for final result

```
# Poll job until SUCCEEDED
curl https://phenom-api.om1.com/v1/job/job_01J90Q2TTM \
  -H "Authorization: Bearer <jwt>"
```

Then fetch results:

```
curl "https://phenom-api.om1.com/v1/job/job_01J90Q2TTM/results?page_size=1000" \
  -H "Authorization: Bearer <jwt>"
```

Example final result payload:


```
{
  "items": [
    {
      "patient_id": "abc123",
      "outcome_id": "v0p3_65plus_fixed_anchor_2023_JanToDec_04012025_any_time_hospitalization_fut
      "probability": 0.7421,
      "prob_upper_95_percent_bound": 0.7934,
      "prob_lower_95_percent_bound": 0.6912,
      "relative_probability": 2.18,
      "rel_upper_95_percent_bound": 2.36,
      "rel_lower_95_percent_bound": 2.02,
      "bin_id": 9,
      "num_bins": 100
    }
  ],
  "next_page_token": null,
  "total_rows": 1
}
```

Quick Start – Batch

End-to-end example: upload data, run a job, fetch results.

1. Create a batch

```
curl -X POST https://phenom-api.om1.com/v1/batch \
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "october_data_load",
    "upload_method": "s3_multipart"
  }'

# Response (example)
{
  "batch_id": "batch_0199bb07-3b87-7e83-8842-ad9d52a3c472",
  "status": "CREATED"
}
```

2. Upload patient data

```
# Create upload session for one CSV
curl -X POST https://phenom-api.om1.com/v1/batch/{batch_id}/upload \
  -H "Authorization: Bearer <jwt>" \
```

```
  -d '{
    "object_type": "patients",
    "filename": "patients_2025_10.csv",
    "mode": "single"
  }'

# Use the presigned URL returned in the response to upload the file.
# Repeat for other object_types: lab_results, diagnosis, procedures, medications.
```

3. Finalize the batch

```
curl -X POST https://phenom-api.om1.com/v1/batch/{batch_id}/finalize \
  -H "Authorization: Bearer <jwt>"

# Response (example)
{
  "batch_id": "batch_0199bb07-3b87-7e83-8842-ad9d52a3c472",
  "status": "FINALIZED",
  "manifest_key": "batches/batch_.../manifest.json"
}
```

4. Start the job

```
curl -X POST https://phenom-api.om1.com/v1/batch/{batch_id}/start \
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '{
    "outcome_ids": [
      "v0p3_65plus_fixed_anchor_2023_JanToDec_04012025_any_time_hospitalization_future_1month"
    ]
  }'

# Response (example)
{
  "job_id": "job_01J90Q2TTM",
  "status": "QUEUED"
}
```

5. Poll job status

```
curl https://phenom-api.om1.com/v1/job/{job_id} \
  -H "Authorization: Bearer <jwt>"

# When complete (example)
{
  "job_id": "job_01J90Q2TTM",
  "status": "SUCCEEDED",
  "metrics": {
    "records_processed": 1245678,
    "patients_scored": 101234,
    "patients_excluded": 1123
  }
}
```

6. Fetch results

```
# Option 1: Paginated API results
curl "https://phenom-api.om1.com/v1/job/{job_id}/results?page_size=1000" \
  -H "Authorization: Bearer <jwt>"

# Option 2: Download full results (manifest with shard URLs)
curl https://phenom-api.om1.com/v1/job/{job_id}/results/download \
  -H "Authorization: Bearer <jwt>"
```

Additional job endpoints are available to inspect **exclusions** and **errors**:

- `/v1/job/{job_id}/exclusions`
- `/v1/job/{job_id}/errors`

Security & Compliance

- **HIPAA & BAA**: HIPAA-compliant infrastructure with BAA coverage
- **Tenant isolation**: All resources are scoped to your tenant via JWT claims
- **Encryption**: SSE-S3 at rest; TLS in transit
- **PHI handling**: PHI appears **only** in uploaded data, not in resource names or metadata