

Analytics Service API Documentation

Base URL

```
http://localhost:8009/api/v1
```

Authentication

All endpoints require JWT authentication unless otherwise specified.

Header:

```
Authorization: Bearer <your-jwt-token>
```

Endpoints Overview

The Analytics Service provides **35 API endpoints** across 7 main categories:

1. **Health & Status** (2 endpoints)
2. **Metrics** (8 endpoints)
3. **Dashboards** (7 endpoints)
4. **Data Sync** (6 endpoints)
5. **Partner Analytics** (4 endpoints)
6. **Project Analytics** (4 endpoints)
7. **Social Media Analytics** (3 endpoints)
8. **Notification Analytics** (3 endpoints)

1. Health & Status

GET /health

Basic health check without dependency verification.

Auth: Not required

Response:

```
{
  "status": "healthy",
  "service": "Analytics Service",
  "version": "0.1.0",
  "timestamp": "2024-12-24T10:00:00Z"
}
```

GET /ready

Readiness check with database connectivity verification.

Auth: Not required

Response:

```
{
  "status": "ready",
  "checks": {
    "database": "connected"
  }
}
```

2. Metrics API

POST /metrics

Create a new metric.

Request Body:

```
{
  "service_name": "partners_crm",
  "metric_type": "donation",
  "metric_name": "total_donations",
  "value": 5000.50,
  "timestamp": "2024-12-24T10:00:00Z",
  "dimensions": {
    "partner_type": "individual",
    "country": "USA"
  },
  "meta": {
    "currency": "USD",
    "campaign_id": "winter-2024"
  }
}
```

Response:

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "service_name": "partners_crm",
  "metric_type": "donation",
  "metric_name": "total_donations",
  "value": 5000.50,
  "timestamp": "2024-12-24T10:00:00Z",
  "date": "2024-12-24",
  "dimensions": {"partner_type": "individual"},
  "meta": {"currency": "USD"},
  "created_at": "2024-12-24T10:00:01Z"
}
```

GET /metrics/{id}

Get a specific metric by ID.

Path Parameters:

- `id` (UUID): Metric ID

Response: Same as POST response

GET /metrics

List metrics with optional filters.

Query Parameters:

- `service_name` (string, optional): Filter by service
- `metric_type` (string, optional): Filter by type
- `metric_name` (string, optional): Filter by name
- `start_date` (date, optional): Start date filter
- `end_date` (date, optional): End date filter
- `skip` (int, default: 0): Pagination offset
- `limit` (int, default: 100): Pagination limit

Response:

```
[  
  {  
    "id": "550e8400-e29b-41d4-a716-446655440000",  
    "service_name": "partners_crm",  
    "metric_name": "total_donations",  
    "value": 5000.50,  
    ...  
  }  
]
```

PATCH /metrics/{id}

Update a metric.

Request Body:

```
{  
  "value": 6000.00,  
  "dimensions": {"updated": true}  
}
```

DELETE /metrics/{id}

Delete a metric.

Response:

```
{  
  "message": "Metric deleted successfully"  
}
```

GET /metrics/aggregate

Aggregate metrics with statistics.

Query Parameters:

- `service_name` (string, optional)
- `metric_type` (string, optional)
- `metric_name` (string, optional)
- `start_date` (date, optional)
- `end_date` (date, optional)
- `group_by` (string[], optional): Fields to group by

Response:

```
{
  "sum": 150000.00,
  "avg": 5000.00,
  "min": 100.00,
  "max": 25000.00,
  "count": 30
}
```

GET /metrics/by-service/{service_name}

Get all metrics for a specific service.

Path Parameters:

- `service_name` (string): Service name

GET /metrics/by-type/{metric_type}

Get all metrics of a specific type.

Path Parameters:

- `metric_type` (string): Metric type

GET /metrics/time-series

Get time-series data for a metric.

Query Parameters:

- `metric_name` (string, required)
- `service_name` (string, optional)
- `start_date` (date, optional)
- `end_date` (date, optional)
- `interval` (string, default: "day"): hour, day, week, month

Response:

```
[
  {
    "timestamp": "2024-12-24",
    "value": 5000.00
  },
  {
    "timestamp": "2024-12-25",
    "value": 5500.00
  }
]
```

3. Dashboards API

POST /dashboards

Create a new dashboard.

Request Body:

```
{
  "name": "Partner Overview",
  "dashboard_type": "partner",
  "description": "Partner performance dashboard",
  "config": {
    "widgets": [
      {
        "type": "metric",
        "metric_name": "total_donations",
        "position": {"x": 0, "y": 0, "w": 4, "h": 2}
      }
    ]
  },
  "created_by": "user-123",
  "is_default": false,
  "is_public": true
}
```

Response:

```
{
  "id": "dashboard-uuid",
  "name": "Partner Overview",
  "dashboard_type": "partner",
  "created_at": "2024-12-24T10:00:00Z",
  ...
}
```

GET /dashboards/{id}

Get a specific dashboard.

GET /dashboards

List dashboards.

Query Parameters:

- `dashboard_type` (string, optional)
- `created_by` (string, optional)
- `is_default` (boolean, optional)
- `is_public` (boolean, optional)

PUT /dashboards/{id}

Update a dashboard.

DELETE /dashboards/{id}

Delete a dashboard.

GET /dashboards/{id}/data

Get dashboard data with populated widget values.

Response:

```
{
  "dashboard_id": "dashboard-uuid",
  "name": "Partner Overview",
  "widgets": [
    {
      "type": "metric",
      "metric_name": "total_donations",
      "data": {
        "value": 150000.00,
        "change": "+12.5%"
      }
    }
  ]
}
```

GET /dashboards/executive

Get the default executive dashboard.

4. Data Sync API

POST /sync

Trigger a data synchronization.

Request Body:

```
{
  "service_name": "partners_crm",
  "sync_type": "incremental"
}
```

Response:

```
{
  "id": "sync-uuid",
  "service_name": "partners_crm",
  "sync_type": "incremental",
  "status": "pending",
  "start_time": "2024-12-24T10:00:00Z"
}
```

GET /sync/{id}

Get sync status.

Response:

```
{
  "id": "sync-uuid",
  "status": "running",
  "records_processed": 1500,
  "records_failed": 3,
  "progress": 75
}
```

GET /sync

List sync records.

Query Parameters:

- `service_name` (string, optional)
- `status` (string, optional)
- `sync_type` (string, optional)

GET /sync/status

Get current sync status for all services.

Query Parameters:

- `service_name` (string, optional): Filter by service

Response:

```
{
  "service_name": "partners_crm",
  "last_sync": "2024-12-24T09:00:00Z",
  "status": "completed",
  "next_sync": "2024-12-24T12:00:00Z"
}
```

GET /sync/statistics

Get sync statistics.

Response:

```
{
  "total_syncs": 150,
  "successful_syncs": 148,
  "failed_syncs": 2,
  "total_records_processed": 50000,
  "avg_sync_duration": 120
}
```

POST /sync/aggregate-all

Trigger aggregation for all services.

Response:

```
{
  "message": "Aggregation triggered for all services",
  "syncs": [
    {"service_name": "partners_crm", "sync_id": "uuid1"},
    {"service_name": "projects", "sync_id": "uuid2"}
  ]
}
```

5. Partner Analytics API

GET /analytics/partners/statistics

Get partner statistics.

Query Parameters:

- `start_date` (date, optional)
- `end_date` (date, optional)

Response:

```
{
  "total_partners": 500,
  "active_partners": 450,
  "new_partners": 25,
  "total_donations": 150000.00,
  "avg_donation": 300.00
}
```

GET /analytics/partners/donations

Get donation trends.

Response:

```
{
  "trends": [
    {"date": "2024-12-24", "amount": 5000.00},
    {"date": "2024-12-25", "amount": 5500.00}
  ],
  "total": 150000.00,
  "growth": "+12.5%"
}
```

GET /analytics/partners/engagement

Get engagement metrics.

Response:

```
{
  "engagement_rate": 75.5,
  "active_rate": 90.0,
  "retention_rate": 85.5
}
```

GET /analytics/partners/breakdown

Get partner type breakdown.

Response:

```
{
  "by_type": {
    "individual": 350,
    "organization": 100,
    "church": 50
  },
  "by_country": {
    "USA": 300,
    "Canada": 100,
    "UK": 50
  }
}
```

6. Project Analytics API

GET /analytics/projects/statistics

Get project statistics.

Response:

```
{
  "total_projects": 50,
  "active_projects": 30,
  "completed_projects": 20,
  "total_beneficiaries": 10000
}
```

GET /analytics/projects/impact

Get impact metrics.

Response:

```
{
  "total_impact_score": 8500,
  "avg_impact_score": 170,
  "beneficiaries_reached": 10000
}
```

GET /analytics/projects/completion

Get completion rates.

Response:

```
{
  "completion_rate": 40.0,
  "on_time_completion": 75.0,
  "avg_duration_days": 180
}
```

GET /analytics/projects/beneficiaries

Get beneficiary metrics.

Response:

```
{
  "total_beneficiaries": 10000,
  "direct_beneficiaries": 5000,
  "indirect_beneficiaries": 5000,
  "by_category": {
    "children": 4000,
    "families": 3000,
    "communities": 3000
  }
}
```

7. Social Media Analytics API

GET /analytics/social-media/performance

Get performance metrics.

Response:

```
{
  "total_posts": 150,
  "total_engagement": 25000,
  "avg_engagement_rate": 5.5,
  "reach": 100000
}
```

GET /analytics/social-media/platforms

Get platform comparison.

Response:

```
{
  "facebook": {
    "posts": 50,
    "engagement": 10000,
    "reach": 50000
  },
  "instagram": {
    "posts": 100,
    "engagement": 15000,
    "reach": 50000
  }
}
```

GET /analytics/social-media/engagement

Get engagement trends.

Response:

```
{
  "trends": [
    {"date": "2024-12-24", "engagement": 500},
    {"date": "2024-12-25", "engagement": 550}
  ],
  "growth_rate": "+10%"
}
```

8. Notification Analytics API

GET /analytics/notifications/statistics

Get notification statistics.

Response:

```
{
  "total_sent": 5000,
  "delivered": 4950,
  "failed": 50,
  "delivery_rate": 99.0
}
```

GET /analytics/notifications/delivery

Get delivery rates.

Response:

```
{
  "by_channel": {
    "email": {"sent": 3000, "delivered": 2970, "rate": 99.0},
    "sms": {"sent": 2000, "delivered": 1980, "rate": 99.0}
  }
}
```

GET /analytics/notifications/channels

Get channel effectiveness.

Response:

```
{
  "email": {"open_rate": 25.0, "click_rate": 5.0},
  "sms": {"response_rate": 15.0},
  "push": {"open_rate": 40.0}
}
```

Enum Values

Service Names

- auth
- content
- partners_crm
- projects
- social_media
- notification

Metric Types

- donation
- partner
- project
- beneficiary
- social_post
- notification
- engagement
- conversion
- revenue

Dashboard Types

- executive
- partner

- project
- social_media
- notification
- custom

Sync Types

- full
- incremental
- manual

Sync Status

- pending
- running
- completed
- failed

Error Responses

All endpoints return consistent error responses:

400 Bad Request

```
{
  "detail": "Invalid request parameters"
}
```

401 Unauthorized

```
{
  "detail": "Not authenticated"
}
```

404 Not Found

```
{
  "detail": "Resource not found"
}
```

500 Internal Server Error

```
{
  "detail": "Internal server error"
}
```

Rate Limiting

API endpoints are rate limited:

- **Authenticated requests:** 1000 requests/hour
 - **Unauthenticated requests:** 100 requests/hour
-

Pagination

List endpoints support pagination:

- `skip` : Offset (default: 0)
 - `limit` : Limit (default: 100, max: 1000)
-

Interactive Documentation

Explore the API interactively:

- **Swagger UI:** <http://localhost:8009/docs>
- **ReDoc:** <http://localhost:8009/redoc>