

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 **60 API endpoints** across 13 main categories:

Phase 1 (35 endpoints)

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)

Phase 2 (25 endpoints)

1. **Reports** (7 endpoints)
2. **Goals** (8 endpoints)
3. **Scheduled Jobs** (6 endpoints)
4. **Advanced Analytics** (8 endpoints)
5. **Data Visualization** (5 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>

Phase 2 Features

The Analytics Service Phase 2 adds **25 new API endpoints** across 5 additional categories:

1. **Reports** (7 endpoints)
2. **Goals** (8 endpoints)
3. **Scheduled Jobs** (6 endpoints)
4. **Advanced Analytics** (8 endpoints)
5. **Data Visualization** (5 endpoints)

Total Endpoints: 60

9. Reports API

POST /reports/generate

Generate a new analytics report.

Request Body:

```
{
  "name": "Monthly Donations Report",
  "description": "Comprehensive donation metrics for the month",
  "report_type": "monthly",
  "format": "pdf",
  "parameters": {
    "service_name": "partners_crm",
    "metric_type": "donation",
    "start_date": "2024-11-01T00:00:00Z",
    "end_date": "2024-11-30T23:59:59Z"
  },
  "created_by": "user-uuid"
}
```

Response:

```
{
  "id": "report-uuid",
  "name": "Monthly Donations Report",
  "report_type": "monthly",
  "format": "pdf",
  "status": "pending",
  "created_at": "2024-12-24T10:00:00Z",
  "created_by": "user-uuid"
}
```

GET /reports/{id}

Retrieve a specific report by ID.

Response:

```
{
  "id": "report-uuid",
  "name": "Monthly Donations Report",
  "report_type": "monthly",
  "format": "pdf",
  "status": "completed",
  "file_path": "/reports/2024-12/monthly-donations.pdf",
  "file_size": 245678,
  "created_at": "2024-12-24T10:00:00Z",
  "completed_at": "2024-12-24T10:05:00Z"
}
```

GET /reports

List all reports with optional filters.

Query Parameters:

- `report_type` (optional): Filter by report type (daily, weekly, monthly, annual, custom)
- `format` (optional): Filter by format (pdf, excel, csv, json)
- `status` (optional): Filter by status (pending, generating, completed, failed)
- `limit` (optional): Number of results (default: 50, max: 100)
- `offset` (optional): Pagination offset

Response:

```
[
  {
    "id": "report-uuid-1",
    "name": "Weekly Summary",
    "report_type": "weekly",
    "format": "excel",
    "status": "completed"
  },
  {
    "id": "report-uuid-2",
    "name": "Annual Review",
    "report_type": "annual",
    "format": "pdf",
    "status": "generating"
  }
]
```

GET /reports/{id}/download

Download a completed report file.

Response: Binary file content with appropriate content-type header

POST /reports/{id}/email

Email a report to specified recipients.

Request Body:

```
{
  "recipients": ["user@example.com", "manager@example.com"],
  "subject": "Monthly Donations Report",
  "message": "Please find attached the monthly donations report."
}
```

Response:

```
{
  "status": "sent",
  "sent_at": "2024-12-24T10:15:00Z",
  "recipients_count": 2
}
```

POST /reports/schedule

Schedule a recurring report.

Request Body:

```
{
  "name": "Daily Summary Report",
  "report_type": "daily",
  "format": "csv",
  "parameters": {},
  "scheduled": true,
  "schedule_config": {
    "cron": "0 8 * * *",
    "timezone": "UTC"
  },
  "recipients": ["team@example.com"],
  "created_by": "user-uuid"
}
```

Response:

```
{
  "id": "report-uuid",
  "name": "Daily Summary Report",
  "scheduled": true,
  "schedule_config": {
    "cron": "0 8 * * *",
    "next_run": "2024-12-25T08:00:00Z"
  }
}
```

DELETE /reports/{id}

Delete a report and its associated file.

Response: 204 No Content**10. Goals API****POST /goals**

Create a new KPI goal.

Request Body:

```
{
  "name": "Q4 Donation Target",
  "description": "Reach $100,000 in donations by end of Q4",
  "metric_type": "donation",
  "target_value": 100000.0,
  "current_value": 25000.0,
  "start_date": "2024-10-01T00:00:00Z",
  "end_date": "2024-12-31T23:59:59Z",
  "alert_threshold": 0.75,
  "created_by": "user-uuid"
}
```

Response:

```
{
  "id": "goal-uuid",
  "name": "Q4 Donation Target",
  "metric_type": "donation",
  "target_value": 100000.0,
  "current_value": 25000.0,
  "progress_percentage": 25.0,
  "status": "active",
  "forecast_value": 90000.0,
  "on_track": false,
  "created_at": "2024-12-24T10:00:00Z"
}
```

GET /goals/{id}

Retrieve a specific goal by ID.

Response:

```
{
  "id": "goal-uuid",
  "name": "Q4 Donation Target",
  "metric_type": "donation",
  "target_value": 100000.0,
  "current_value": 45000.0,
  "progress_percentage": 45.0,
  "status": "active",
  "days_remaining": 7,
  "daily_required_rate": 7857.14
}
```

GET /goals

List all goals with optional filters.

Query Parameters:

- `metric_type` (optional): Filter by metric type
- `status` (optional): Filter by status (active, achieved, failed, cancelled)
- `limit` (optional): Number of results
- `offset` (optional): Pagination offset

Response:

```
[
  {
    "id": "goal-uuid-1",
    "name": "Q4 Donation Target",
    "metric_type": "donation",
    "progress_percentage": 45.0,
    "status": "active"
  },
  {
    "id": "goal-uuid-2",
    "name": "Partner Growth",
    "metric_type": "partner",
    "progress_percentage": 95.0,
    "status": "active"
  }
]
```

PUT /goals/{id}

Update a goal.

Request Body:

```
{
  "target_value": 120000.0,
  "description": "Updated target to $120,000"
}
```

Response:

```
{
  "id": "goal-uuid",
  "target_value": 120000.0,
  "progress_percentage": 37.5,
  "updated_at": "2024-12-24T10:30:00Z"
}
```

POST /goals/{id}/update-progress

Update the current progress of a goal.

Request Body:

```
{
  "current_value": 55000.0
}
```

Response:

```
{
  "id": "goal-uuid",
  "current_value": 55000.0,
  "progress_percentage": 55.0,
  "status": "active",
  "on_track": true,
  "updated_at": "2024-12-24T11:00:00Z"
}
```

GET /goals/{id}/forecast

Get forecast for goal achievement.

Response:

```
{
  "goal_id": "goal-uuid",
  "forecast_value": 98000.0,
  "confidence": 0.85,
  "on_track": false,
  "days_remaining": 7,
  "daily_required_rate": 6428.57,
  "current_daily_rate": 1000.0,
  "projected_completion_date": "2025-01-15T00:00:00Z"
}
```

GET /goals/forecast

Get forecasts for all active goals.

Response:

```
[
  {
    "goal_id": "goal-uuid-1",
    "goal_name": "Q4 Donation Target",
    "forecast_value": 98000.0,
    "on_track": false
  },
  {
    "goal_id": "goal-uuid-2",
    "goal_name": "Partner Growth",
    "forecast_value": 52.0,
    "on_track": true
  }
]
```

DELETE /goals/{id}

Delete a goal.

Response: 204 No Content

11. Scheduled Jobs API

POST /scheduled-jobs

Create a new scheduled job.

Request Body:

```
{
  "name": "Daily Data Sync",
  "description": "Sync all service data daily",
  "job_type": "data_sync",
  "cron_expression": "0 2 * * *",
  "config": {
    "services": ["partners_crm", "projects", "social_media"],
    "sync_type": "incremental"
  },
  "is_active": true
}
```

Response:

```
{
  "id": "job-uuid",
  "name": "Daily Data Sync",
  "job_type": "data_sync",
  "cron_expression": "0 2 * * *",
  "is_active": true,
  "next_run": "2024-12-25T02:00:00Z",
  "created_at": "2024-12-24T10:00:00Z"
}
```

GET /scheduled-jobs/{id}

Retrieve a specific scheduled job.

Response:

```
{
  "id": "job-uuid",
  "name": "Daily Data Sync",
  "job_type": "data_sync",
  "cron_expression": "0 2 * * *",
  "is_active": true,
  "last_run": "2024-12-24T02:00:00Z",
  "next_run": "2024-12-25T02:00:00Z",
  "last_status": "success",
  "total_executions": 45,
  "successful_executions": 43,
  "failed_executions": 2
}
```

GET /scheduled-jobs

List all scheduled jobs.

Query Parameters:

- `job_type` (optional): Filter by type (data_sync, report_generation, goal_update, custom)
- `is_active` (optional): Filter by active status (true/false)

Response:

```
[
  {
    "id": "job-uuid-1",
    "name": "Daily Data Sync",
    "job_type": "data_sync",
    "is_active": true,
    "next_run": "2024-12-25T02:00:00Z"
  },
  {
    "id": "job-uuid-2",
    "name": "Weekly Report",
    "job_type": "report_generation",
    "is_active": true,
    "next_run": "2024-12-30T09:00:00Z"
  }
]
```

PUT /scheduled-jobs/{id}

Update a scheduled job.

Request Body:

```
{
  "cron_expression": "0 3 * * *",
  "is_active": false
}
```

Response:

```
{
  "id": "job-uuid",
  "cron_expression": "0 3 * * *",
  "is_active": false,
  "updated_at": "2024-12-24T10:30:00Z"
}
```

POST /scheduled-jobs/{id}/trigger

Manually trigger a job execution.

Request Body (optional):

```
{
  "config_override": {
    "priority": "high",
    "services": ["partners_crm"]
  }
}
```

Response:

```
{
  "job_id": "job-uuid",
  "execution_id": "execution-uuid",
  "status": "running",
  "started_at": "2024-12-24T10:45:00Z"
}
```

GET /scheduled-jobs/{id}/stats

Get execution statistics for a job.

Response:

```
{
  "job_id": "job-uuid",
  "total_executions": 45,
  "successful_executions": 43,
  "failed_executions": 2,
  "success_rate": 95.56,
  "average_duration": 245.5,
  "last_execution": {
    "started_at": "2024-12-24T02:00:00Z",
    "completed_at": "2024-12-24T02:04:15Z",
    "status": "success",
    "duration": 255
  }
}
```

DELETE /scheduled-jobs/{id}

Delete a scheduled job.

Response: 204 No Content

12. Advanced Analytics API

GET /analytics/advanced/predictions

Get predictive analytics for metrics.

Query Parameters:

- `service_name` : Service to analyze
- `metric_type` : Metric type
- `days` (optional): Days to predict (default: 30)
- `confidence` (optional): Confidence level (0.80, 0.90, 0.95)

Response:

```
{
  "service_name": "partners_crm",
  "metric_type": "donation",
  "predictions": [
    {
      "date": "2024-12-25",
      "predicted_value": 5250.0,
      "confidence_lower": 4800.0,
      "confidence_upper": 5700.0
    },
    {
      "date": "2024-12-26",
      "predicted_value": 5300.0,
      "confidence_lower": 4850.0,
      "confidence_upper": 5750.0
    }
  ],
  "trend": "increasing",
  "confidence": 0.95
}
```

GET /analytics/advanced/forecasts

Get time-series forecasts with seasonality.

Query Parameters:

- `service_name` : Service to analyze
- `metric_type` : Metric type
- `start_date` : Historical start date
- `end_date` : Historical end date
- `forecast_days` : Days to forecast

Response:

```
{
  "historical": [
    {"date": "2024-12-01", "value": 4500.0},
    {"date": "2024-12-02", "value": 4750.0}
  ],
  "forecast": [
    {"date": "2024-12-25", "value": 5200.0},
    {"date": "2024-12-26", "value": 5300.0}
  ],
  "seasonality": {
    "detected": true,
    "pattern": "weekly",
    "strength": 0.65
  },
  "confidence": 0.90
}
```

GET /analytics/advanced/comparisons

Compare metrics across time periods.

Query Parameters:

- `metric_type` : Metric type to compare
- `start_date` : Current period start

- `end_date` : Current period end
- `compare_to_previous_period` (optional): Boolean

Response:

```
{
  "current_period": {
    "start_date": "2024-12-01",
    "end_date": "2024-12-24",
    "total_value": 125000.0,
    "average_value": 5208.33,
    "count": 24
  },
  "previous_period": {
    "start_date": "2024-11-01",
    "end_date": "2024-11-24",
    "total_value": 110000.0,
    "average_value": 4583.33,
    "count": 24
  },
  "change_percentage": 13.64,
  "change_absolute": 15000.0,
  "trend": "positive"
}
```

POST /analytics/advanced/custom-calculations

Perform custom calculations on metrics.

Request Body:

```
{
  "calculation_type": "average",
  "service_name": "social_media",
  "metric_type": "engagement",
  "start_date": "2024-12-01T00:00:00Z",
  "end_date": "2024-12-24T23:59:59Z",
  "group_by": "date",
  "filters": {
    "platform": "instagram"
  }
}
```

Response:

```
{
  "calculation_type": "average",
  "result": [
    {"date": "2024-12-01", "value": 450.5},
    {"date": "2024-12-02", "value": 475.2}
  ],
  "summary": {
    "overall_average": 462.5,
    "total_count": 24
  }
}
```

GET /analytics/advanced/anomalies

Detect anomalies in metric data.

Query Parameters:

- `service_name` : Service to analyze
- `metric_type` : Metric type
- `start_date` : Analysis start date
- `end_date` : Analysis end date
- `sensitivity` (optional): Detection sensitivity (0.90-0.99)

Response:

```
{
  "anomalies": [
    {
      "date": "2024-12-15",
      "value": 8500.0,
      "expected_value": 5200.0,
      "deviation": 63.46,
      "severity": "high"
    }
  ],
  "threshold": {
    "upper": 6500.0,
    "lower": 3500.0
  },
  "sensitivity": 0.95
}
```

GET /analytics/advanced/trends

Analyze metric trends.

Query Parameters:

- `service_name` : Service to analyze
- `metric_type` : Metric type
- `start_date` : Analysis start date
- `end_date` : Analysis end date
- `interval` (optional): Aggregation interval (day, week, month)

Response:

```
{
  "trend": "increasing",
  "direction": "up",
  "strength": 0.78,
  "slope": 125.5,
  "data_points": 90,
  "r_squared": 0.82,
  "interpretation": "Strong upward trend detected"
}
```

POST /analytics/advanced/correlations

Analyze correlation between metrics.

Request Body:

```
{
  "metric1": {
    "service_name": "partners_crm",
    "metric_type": "donation"
  },
  "metric2": {
    "service_name": "social_media",
    "metric_type": "engagement"
  },
  "start_date": "2024-09-01T00:00:00Z",
  "end_date": "2024-12-24T23:59:59Z"
}
```

Response:

```
{
  "correlation": 0.67,
  "strength": "moderate",
  "p_value": 0.001,
  "significant": true,
  "interpretation": "Moderate positive correlation detected"
}
```

GET /analytics/advanced/benchmarks

Get performance benchmarks.

Query Parameters:

- metric_type : Metric type to benchmark
- period (optional): Time period (day, week, month, year)

Response:

```
{
  "metric_type": "conversion",
  "period": "month",
  "average": 3.45,
  "median": 3.20,
  "percentiles": {
    "p25": 2.80,
    "p50": 3.20,
    "p75": 4.10,
    "p90": 5.20,
    "p95": 6.50
  },
  "best": 8.90,
  "worst": 1.20,
  "data_points": 12
}
```

13. Data Visualization API

GET /analytics/visualizations/line-chart

Get line chart data.

Query Parameters:

- `service_name` : Service to visualize
- `metric_type` : Metric type
- `start_date` : Chart start date
- `end_date` : Chart end date
- `interval` (optional): Data interval (hour, day, week, month)

Response:

```
{
  "chart_type": "line",
  "labels": ["2024-12-01", "2024-12-02", "2024-12-03"],
  "datasets": [
    {
      "label": "Donations",
      "data": [5000, 5200, 4800],
      "color": "#4CAF50"
    }
  ]
}
```

GET /analytics/visualizations/bar-chart

Get bar chart data with aggregation.

Query Parameters:

- `service_name` : Service to visualize
- `metric_type` : Metric type
- `start_date` : Chart start date
- `end_date` : Chart end date
- `group_by` : Grouping dimension

Response:

```
{
  "chart_type": "bar",
  "labels": ["USA", "Canada", "UK", "Germany"],
  "datasets": [
    {
      "label": "Donations by Country",
      "data": [45000, 22000, 18000, 15000],
      "colors": ["#4CAF50", "#2196F3", "#FFC107", "#9C27B0"]
    }
  ]
}
```

GET /analytics/visualizations/pie-chart

Get pie chart distribution data.

Query Parameters:

- `service_name` : Service to visualize
- `metric_type` : Metric type
- `dimension` : Dimension to distribute by
- `start_date` : Data start date
- `end_date` : Data end date

Response:

```
{
  "chart_type": "pie",
  "labels": ["Individual", "Corporate", "Foundation", "Government"],
  "data": [45.5, 30.2, 18.8, 5.5],
  "colors": ["#4CAF50", "#2196F3", "#FFC107", "#9C27B0"],
  "total": 100.0
}
```

GET /analytics/visualizations/area-chart

Get area chart data with multiple series.

Query Parameters:

- `service_names` : Comma-separated services
- `metric_type` : Metric type
- `start_date` : Chart start date
- `end_date` : Chart end date
- `interval` (optional): Data interval

Response:

```
{
  "chart_type": "area",
  "labels": ["2024-12-01", "2024-12-02", "2024-12-03"],
  "datasets": [
    {
      "label": "Partners CRM",
      "data": [5000, 5200, 4800],
      "color": "#4CAF50"
    },
    {
      "label": "Projects",
      "data": [3000, 3200, 3100],
      "color": "#2196F3"
    }
  ]
}
```

GET /analytics/visualizations/heatmap

Get heatmap data showing patterns.

Query Parameters:

- `service_name` : Service to visualize
- `metric_type` : Metric type
- `start_date` : Data start date
- `end_date` : Data end date

- `x_dimension` : X-axis dimension (e.g., "hour", "day_of_week")
- `y_dimension` : Y-axis dimension (e.g., "day", "week")

Response:

```
{
  "chart_type": "heatmap",
  "x_labels": ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"],
  "y_labels": ["00:00", "06:00", "12:00", "18:00"],
  "data": [
    [10, 15, 20, 25, 30, 35, 40],
    [45, 50, 55, 60, 65, 70, 75],
    [80, 85, 90, 95, 100, 105, 110],
    [70, 75, 80, 85, 90, 95, 100]
  ],
  "color_scale": {
    "min": 10,
    "max": 110,
    "colors": ["#FFEBEE", "#EF5350", "#C62828"]
  }
}
```

Enum Values

Report Types

- `daily`
- `weekly`
- `monthly`
- `annual`
- `custom`

Report Formats

- `pdf`
- `excel`
- `csv`
- `json`

Report Status

- `pending`
- `generating`
- `completed`
- `failed`

Goal Metric Types

- `donation`
- `partner`
- `project`
- `beneficiary`
- `social_post`

- notification
- engagement
- conversion
- revenue

Goal Status

- active
- achieved
- failed
- cancelled

Job Types

- data_sync
- report_generation
- goal_update
- custom

Job Status

- success
- failed
- running
- pending

Rate Limiting

All endpoints are subject to rate limiting:

- **Authenticated requests:** 1000 requests per hour per user
- **Report generation:** 50 requests per hour per user
- **Scheduled jobs:** 100 requests per hour per user

Rate limit headers included in responses:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 995
X-RateLimit-Reset: 1703419200
```

Error Responses

All Phase 2 endpoints follow the same error response format:

```
{
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid report format",
    "details": {
      "field": "format",
      "allowed_values": ["pdf", "excel", "csv", "json"]
    }
  }
}
```

Common error codes:

- VALIDATION_ERROR (400)
- UNAUTHORIZED (401)
- FORBIDDEN (403)
- NOT_FOUND (404)
- RATE_LIMIT_EXCEEDED (429)
- INTERNAL_ERROR (500)

Pagination

All list endpoints support pagination:

Query Parameters:

- limit : Number of results (default: 50, max: 100)
- offset : Starting position (default: 0)

Response Headers:

```
X-Total-Count: 245
X-Limit: 50
X-Offset: 0
Link: </api/v1/reports?limit=50&offset=50>; rel="next"
```

Last Updated: December 24, 2024 - Phase 2 Complete