# PROJECT DOCUMENTATION

## Super App Platform Infrastructure — Analytics & Logs API

## 1. Overview

The Analytics & Logs API is a foundational infrastructure service of the Super App, designed to provide administrators with actionable insights and transparent records of platform activity. It integrates two critical functions:

• Analytics: Aggregate usage data (e.g., disputes resolved, flags handled) and feed dashboards with trends and charts.

• Audit Logs: Maintain an immutable record of all system and admin actions for compliance, governance, and investigations.

By combining monitoring, reporting, and accountability, this module ensures the Super App remains transparent, trustworthy, and compliant with industry standards.

## 2. Purpose

The purpose of the Analytics & Logs API is to:

1. Enable Visibility Across Services — Provide admins with charts and usage metrics spanning disputes, flags, and platform actions.

2. Ensure Compliance and Governance — Maintain secure, tamper-proof audit logs and support audits (e.g., GDPR, SOC 2).

3. Support Operational Efficiency — Give administrators consolidated activity views and data-driven insights.

## 3. Scope (MVP)

### 3.1 In-Scope Features

• Aggregation of audit data for admin dashboards.

• Statistics for disputes, flags, and system actions.

• Retrieval of audit logs via secure endpoints.

### 3.2 Future Enhancements

• Real-time data streaming to dashboards.

• Advanced reporting (trends, anomalies, predictive analytics).

• Integration with external BI and monitoring systems.

• Cross-service analytics across payments, commerce, and social modules.

## 4. Functional Requirements

• Query the audit\_logs table to generate statistics.

• Return metrics including disputes resolved and flags handled.

• Expose APIs to retrieve raw audit logs with filters.

• Enforce admin-only access with authentication & authorization.

• Return consistent error codes (401 Unauthorized, 403 Forbidden, 404 Not Found).

## 5. Data Model

|  |  |  |
| --- | --- | --- |
| Column | Type | Description |
| id (PK) | INT | Unique identifier for log entry. |
| action | TEXT | Action performed (e.g., dispute\_resolved). |
| user\_id (FK) | INT | ID of the user/admin performing the action. |
| target\_id | INT | Entity affected (dispute ID, flag ID, etc.). |
| created\_at | TIMESTAMP | Time when the action was recorded. |

## 6. API Endpoints

### 6.1 GET /api/v1/admin/analytics

Purpose: Return aggregated platform statistics.

Example Response:  
{  
 "disputes\_resolved": 120,  
 "flags\_handled": 85,  
 "total\_logs": 2500  
}

### 6.2 GET /api/v1/admin/audit-logs

Purpose: Retrieve audit logs with optional filters.

Example Response:  
[  
 {  
 "id": 101,  
 "action": "dispute\_resolved",  
 "user\_id": 9001,  
 "target\_id": 202,  
 "created\_at": "2025-09-15T10:00:00Z"  
 }  
]

## 7. Sequence Flow

1. A user or admin performs an action in the Super App.

2. The action is stored in the audit\_logs table.

3. The Analytics API aggregates data from logs.

4. Admin dashboard consumes APIs to display charts and logs.

## 8. Testing Plan

• Unit Tests — Validate aggregation logic for disputes and flags.

• Integration Tests — Ensure API endpoints return expected values to the dashboard.

• Edge Case Tests — Invalid filter → 400 Bad Request; Unauthorized request → 403 Forbidden; Missing record → 404 Not Found.

## 9. Deliverables

• Fully implemented Analytics & Logs API endpoints.

• Backend logic for chart generation.

• Unit + integration test suite.

• Documentation (API reference, sequence diagrams, compliance notes).

• Integration with Super App admin dashboard.

## 10. Security & Compliance

• Role-Based Access Control (RBAC) — Only authorized admins can query analytics or logs.

• Immutable Audit Logs — All logs are append-only to prevent tampering.

• Data Protection — Sensitive data minimized, encrypted at rest, and masked in reports.

• Compliance Alignment — Supports GDPR, SOC 2, and internal governance standards.