Checklist for Logs

The Logs category tracks system actions, workflows, tasks, and security events, ensuring transparency, debugging, and compliance. This checklist ensures the logging system is granular, secure, and well-organized.

---

1. Structure

[ ] Log Categories:

Separate logs into the following categories:

Workflow Logs: Tracks task creation, assignment, and escalations.

Task Logs: Tracks task execution and updates.

System Logs: Records system-level actions (e.g., startup, configuration changes).

Security Logs: Tracks access attempts, sensitive data access, and security alerts.

Email Logs: Records email sending and receiving events.

[ ] Retention:

Define retention policies for each log type (e.g., 6 months for activity logs, 1 year for security logs).

---

2. Workflow Logs

[ ] Activity Tracking:

Log every action in a workflow (e.g., task creation, routing, completion).

[ ] Escalation Events:

Record escalations, including the reason and recipient.

[ ] Performance Metrics:

Log execution times for workflows and identify bottlenecks.

Example Workflow Log Entry

[2024-11-25 10:00:00] Workflow: Maintenance | Task ID: 123 | Status: Assigned to john.doe@organization.com

---

3. Task Logs

[ ] Lifecycle States:

Log every state change (pending, in-progress, completed).

[ ] Notifications:

Record notifications sent for task updates or escalations.

[ ] Error Tracking:

Log errors related to task execution (e.g., failure to update status).

Example Task Log Entry

[2024-11-25 10:15:00] Task ID: 124 | Status: Completed | Duration: 2 hours

---

4. System Logs

[ ] System Events:

Log startup, shutdown, configuration changes, and deployment actions.

[ ] Performance Metrics:

Record CPU, memory, and disk usage.

[ ] Error Tracking:

Log system-level errors (e.g., failed deployments, missing dependencies).

Example System Log Entry

[2024-11-25 10:30:00] Event: System Startup | Version: 1.2.3 | Environment: Production

---

5. Security Logs

[ ] Access Control:

Log successful and failed login attempts, including IP addresses.

[ ] Sensitive Actions:

Record actions involving sensitive data (e.g., anonymization, user role changes).

[ ] Alert Logs:

Log all security alerts and their resolution status.

Example Security Log Entry

[2024-11-25 10:45:00] Alert: Unauthorized Access Attempt | User: admin | IP: 192.168.1.100 | Status: Blocked

---

6. Email Logs

[ ] Incoming Emails:

Log metadata for all received emails (e.g., sender, subject, timestamp).

[ ] Outgoing Emails:

Log metadata for all sent emails (e.g., recipient, subject, delivery status).

[ ] Error Tracking:

Record errors during email sending/receiving (e.g., connection issues).

Example Email Log Entry

[2024-11-25 11:00:00] Email Sent | To: support@organization.com | Subject: Task Update | Status: Delivered

---

7. Log Security

[ ] Encryption:

Encrypt logs containing sensitive data (e.g., security logs).

[ ] Access Control:

Restrict access to logs based on user roles (e.g., only admins can access security logs).

[ ] Audit Trails:

Maintain a history of log access and modifications.

---

8. Retention Policies

[ ] Log Rotation:

Automatically rotate logs based on size or age (e.g., weekly rotation for high-frequency logs).

[ ] Archival:

Archive logs exceeding the retention period for compliance or auditing.

[ ] Deletion:

Automatically delete logs after the defined retention period.

Example Retention Policy

Workflow Logs: 6 months

Task Logs: 1 year

System Logs: 6 months

Security Logs: 2 years

---

9. Validation

[ ] Log Format:

Ensure all logs follow a consistent format (e.g., timestamps, identifiers).

[ ] Required Fields:

Validate that every log entry includes mandatory fields (e.g., timestamp, category, message).

[ ] Error Handling:

Log any issues encountered while writing logs (e.g., disk full errors).

---

10. Testing

[ ] Functional Tests:

Test that all logging modules (e.g., workflow, task) create expected log entries.

[ ] Edge Cases:

Simulate scenarios with missing or invalid data to ensure logs are still generated correctly.

[ ] Performance:

Test the logging system under high load to ensure it doesn’t slow down the application.

---

Example Application for Logs

Directory Structure

/logs/

├── workflow/

│ ├── workflow\_activity.log

│ ├── escalation\_tracker.log

│ └── workflow\_performance.log

├── tasks/

│ ├── task\_execution.log

│ ├── task\_notifications.log

│ └── overdue\_tasks.log

├── system/

│ ├── system\_activity.log

│ ├── deployment\_actions.log

│ └── performance\_monitor.log

├── security/

│ ├── access\_control.log

│ ├── data\_anonymization.log

│ └── alerts.log

├── email/

│ ├── email\_incoming.log

│ ├── email\_outgoing.log

│ └── email\_errors.log

Checklist Applied

[ ] Workflow Logs:

workflow\_activity.log tracks task assignments and routing.

[ ] Security Logs:

access\_control.log logs all login attempts with timestamps and IP addresses.

[ ] Retention:

Logs are rotated weekly, and old logs are archived after 6 months.

---

Optimized Workflow

1. Define Log Categories:

Start with essential categories (e.g., workflow, system, security).

2. Implement Logging Modules:

Write modular logging functions for each category.

3. Test Logging:

Validate that logs are generated correctly under normal and edge-case scenarios.

4. Enforce Retention Policies:

Automate log rotation and archival to maintain compliance.

---

This checklist ensures Orgo’s logging system is granular, secure, and compliant with retention policies. Let me know if you'd like to implement logging for a specific category (e.g., workflows or security)!