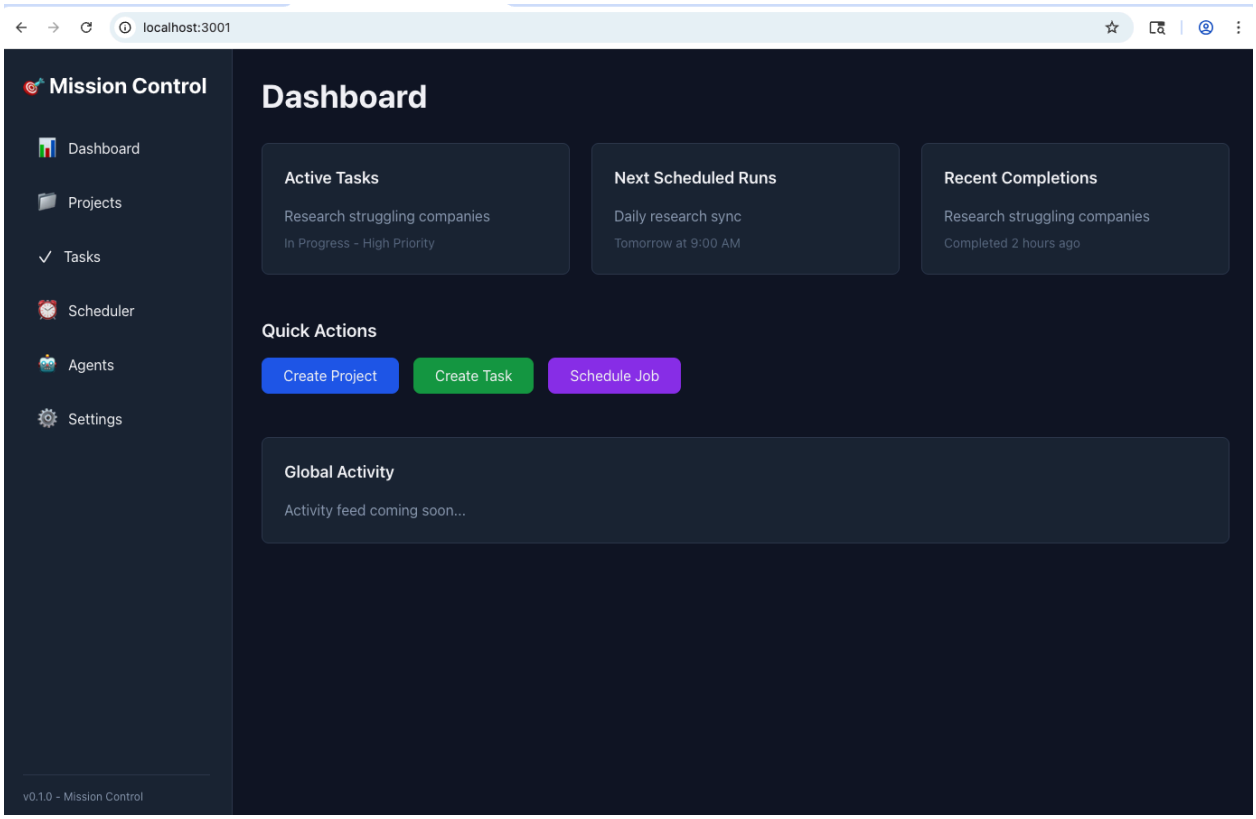
MISSION CONTROL



Overview

OpenClaw Mission Control is your local AI operations dashboard.

It runs at:
http://localhost:3001/
Mission Control does **not** replace the main OpenClaw Web UI.
It is an orchestration layer built for:

- Managing multiple projects
- Structuring execution
- Running agents intentionally
- Scheduling automation
- Tracking outputs and logs
- Operating like a system, not chaos

If OpenClaw is the engine, Mission Control is the command center.

1. Projects

Projects are the foundation of Mission Control.
Every task, agent, schedule, and output is tied to a project.

Creating a Project

When creating a new project, define:

- Project name
- Description
- Project root directory
- Output directory
- Assigned agent(s)
- Optional Git reference

Keep each project isolated.
This prevents cross contamination between builds.

Project Dashboard

Each project includes:

- Active tasks
- Assigned agents
- Scheduled jobs
- Recent runs
- Output artifacts
- Activity timeline

Use this page as your execution overview.

2. Tasks

Tasks are structured work units inside a project.
There are three types:

- Plan
- Build
- Ops

Creating a Task

Each task includes:

- Title
- Description
- Type
- Priority
- Status
- Assigned agent
- Expected deliverables

Tasks move across statuses:
Backlog → Planned → In Progress → Blocked → Done
Use this to manage real execution flow.

Running a Task

Click “Run Task” to trigger execution.
What happens:

1. Mission Control calls the OpenClaw adapter
2. The correct agent is executed
3. Logs stream live
4. Outputs are written to the project output directory
5. Run history is saved

You can monitor:

- Status
- Start time
- End time
- Logs
- Produced artifacts

This is real execution, not simulation.

3. Agents and Sub Agents

Agents are automatically discovered from:
~/.openclaw/openclaw.json
You can:

- View available agents
- Assign agents to projects
- Define a primary agent
- Use sub agents for specialization

Think of this as your AI team structure.
Each project can have a different loadout.

4. Scheduler and Cron Jobs

Mission Control supports recurring automation.

Creating a Scheduled Job

You define:

- Project
- Job type
- Cron expression
- Enabled or disabled state

You can:

- Run immediately
- Enable or disable
- View next run time
- Review run history

The system prevents overlapping runs by default.
Use scheduler jobs for:

- Reporting
- Monitoring
- Outreach
- Maintenance tasks
- Recurring build checks

5. Web Chat (Project Context)

Each project includes a contextual chat interface.
This is not generic prompting.
This is project aware AI.

How Chat Works

1. Select project
2. Select agent
3. Optionally attach:
    - A task
    - Specific files
4. Send message
5. Response streams live

You can:

- Save chat output to task notes
- Create a task from chat
- Export the session to markdown
- Store full chat history

All chat sessions are saved per project.

6. Files and Outputs

Each project includes a file browser for its output directory.
You can:

- Browse directory tree
- Search filenames
- Preview small text files
- Copy file paths

- Open folder in your OS
- Link files to tasks

Artifacts from runs are tracked and organized.
No more digging through random folders.

7. Dashboard Overview

The global Mission Control dashboard shows:

- Active tasks across projects
- Running executions
- Upcoming scheduled jobs
- Recent completions
- Blocked tasks

Use this page to understand system health.

8. Execution Architecture

When a task runs:

1. UI triggers backend API
2. Backend validates project paths
3. OpenClaw CLI is executed via allowlisted command
4. Logs stream via WebSocket
5. Run is persisted in database
6. Output artifacts are indexed

Everything is local.
Everything is tracked.

9. Security Model

Mission Control:

- Binds to 127.0.0.1 by default
- Validates project root paths
- Prevents arbitrary command execution
- Does not store secrets in plain text
- Uses environment variable references where needed

Execution is controlled and bounded.

10. Operational Best Practices

To get the most out of Mission Control:

- Keep projects isolated
- Define clear task deliverables
- Use scheduler for recurring work

- Assign agents intentionally
- Review logs after major runs
- Keep output directories clean and structured

Treat this as infrastructure, not a toy dashboard.

11. What Mission Control Enables

You now have:

- Structured multi project management
- Agent orchestration
- Real execution logging
- Recurring automation
- Contextual AI chat
- Artifact traceability
- System level visibility

This is how you move from prompting to operating.