

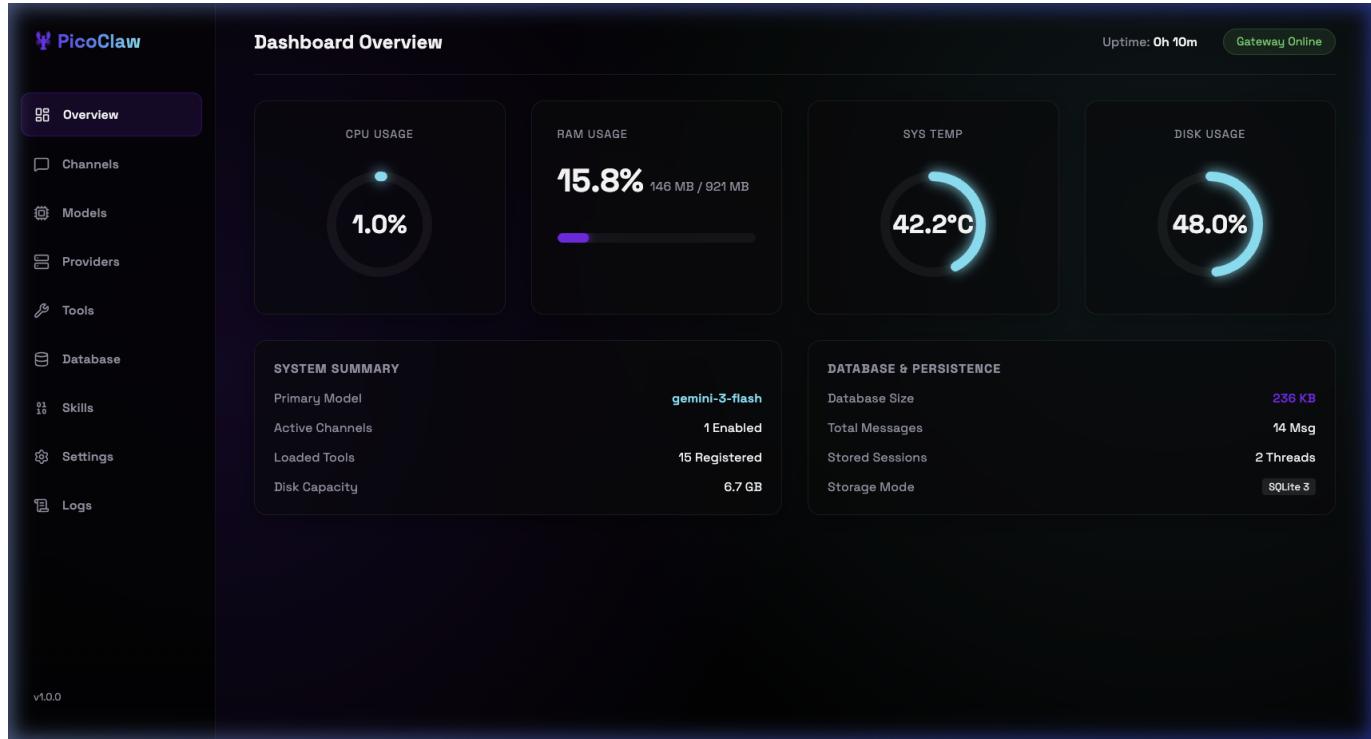
# PicoClaw Dashboard Documentation

## Overview

The PicoClaw Dashboard is a web-based management interface for the PicoClaw personal AI assistant. It provides a centralized view of the system status, configuration, and extensibility options.

## Dashboard Pages

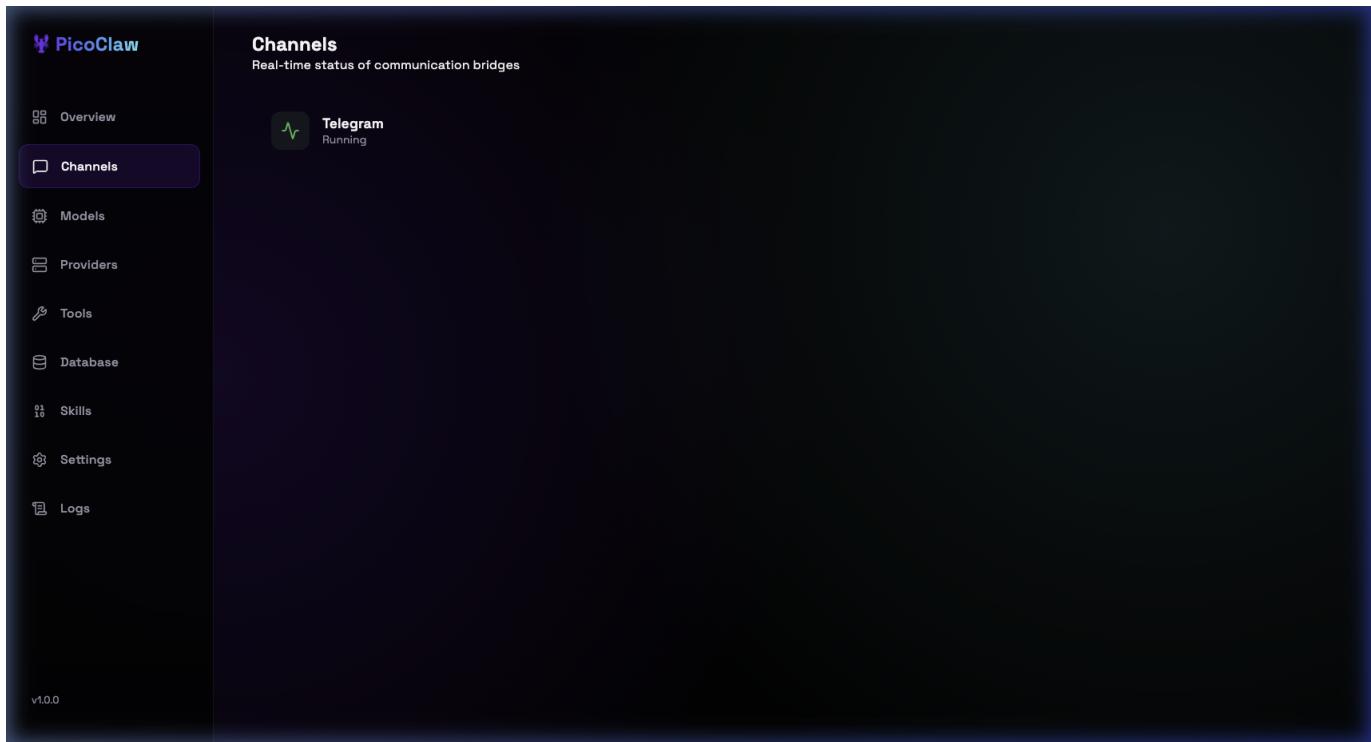
### 1. Overview



The landing page of the dashboard. It displays real-time system metrics including:

- \*\*System Stats\*\*: Uptime, CPU Usage, RAM Usage, and Temperature.
- \*\*Disk Usage\*\*: Total, used, and percentage usage.
- \*\*Database Metrics\*\*: Total message count, session count, and database file size.
- \*\*Metadata\*\*: The primary AI model currently in use and the count of active channels and tools.

### 2. Channels



Displays the status of communication channels.

- - **Active Channels**: Lists channels like Telegram, indicating if they are running.
- - **Connection Status**: Real-time status of the bot connections.

### 3. Models

The screenshot shows the PicoClaw application interface. On the left is a sidebar with the following navigation items:

- Overview
- Channels
- Models** (highlighted)
- Providers
- Tools
- Database
- Skills
- Settings
- Logs

At the bottom of the sidebar, it says "v1.0.0".

The main content area is titled "Models" and subtitle "Configured LLMs and their protocols". It lists five models:

- gpt4** (openai/gpt-4o)  
API Key: .....  
Base: <https://openrouter.ai/api/v1>
- claude-sonnet-4.6** (anthropic/claude-sonnet-4.6)  
API Key: .....  
Base: <https://api.anthropic.com/v1>
- gemini-flash** (antigravity/gemini-3-flash)  
API Key: Not configured  
Base: <https://api.anthropic.com/v1>
- loadbalanced-gpt4** (openai/gpt-5.2)  
API Key: .....  
Base: <https://api1.example.com/v1>
- loadbalanced-gpt4** (openai/gpt-5.2)  
API Key: .....  
Base: <https://api2.example.com/v1>

Lists all available Large Language Models (LLMs) configured in the system.

- - Shows the model name, provider, and parameters.

## 4. Providers

The screenshot shows the PicoClaw application interface. On the left is a sidebar with the following menu items:

- Overview
- Channels
- Models
- Providers** (highlighted)
- Tools
- Database
- Skills
- Settings
- Logs

At the bottom of the sidebar, it says "v1.0.0".

The main content area is titled "Providers" and subtitle "Configured Legacy LLM Providers". It lists six providers with their status:

| Provider   | Status     |
|------------|------------|
| Openrouter | Configured |
| Groq       | Configured |
| Zhipu      | Configured |
| Nvidia     | Configured |
| Ollama     | Configured |
| Moonshot   | Configured |
| Owen       | Configured |

Below the provider list, it says "Base: <http://localhost:11434/v1>".

Displays the configuration of LLM providers (e.g., Gemini, OpenAI, Claude).

- It allows verifying provider connectivity and base API URLs.

## 5. Tools

The screenshot shows the PicoClaw application interface. On the left is a sidebar with the following navigation items:

- Overview
- Channels
- Models
- Providers
- Tools** (highlighted with a blue border)
- Database
- Skills
- Settings
- Logs

On the right, under the "Tools" section, there is a heading "Tools" and a subtitle "Agent capabilities and schemas". Below this, three tool entries are listed:

- read\_file**: Described as "Read the contents of a file".

```
{  "properties": {    "path": {      "description": "Path to the file to read",      "type": "string"    }  },  "required": [    "path"  ],  "type": "object"
```
- list\_dir**: Described as "List files and directories in a path".

```
{  "properties": {    "path": {      "description": "Path to list",      "type": "string"    }  },  "required": [    "path"  ],  "type": "object"
```
- i2c**: Described as "Interact with I2C bus devices for reading sensors and controlling peripherals. Actions: detect (list buses), scan (find devices on a bus), read (read bytes from device), write (send bytes to device). Linux only."

```
Actions: detect (list buses), scan (find devices on a bus), read (read bytes from device), write (send bytes to device). Linux only.
```

A library of all tools currently registered with the agent.

- - Each tool entry includes its name, description, and the parameters it accepts.

## 6. Database

The screenshot shows the PicoClaw application interface. On the left is a sidebar with the following navigation items:

- Overview
- Channels
- Models
- Providers
- Tools
- Database** (selected)
- Skills
- Settings
- Logs

At the bottom of the sidebar, it says "v1.0.0".

The main area is titled "Database Explorer" and has the sub-instruction "Raw access to local SQLite persistence". It features a table with the following data:

| KEY   | UPDATED_AT                          | VALUE   | ACTIONS            |
|-------|-------------------------------------|---|--------------------|
| state | 2026-02-24T16:40:25.945689281+08:00 | {"last_channel": "telegram:1851490538", "timestamp": "2026-02-24..."} | <span>trash</span> |

At the top right of the main area, there are buttons for "global\_state" (dropdown), "refresh" (blue circle), and "Wipe Table" (triangle icon).

Allows direct interaction with the underlying SQLite database.

- - **Tables**: List all tables in the system.
- - **Query/View**: Browse rows within tables.
- - **Maintenance**: Delete specific rows or wipe entire tables for cleanup.

## 7. Skills

The screenshot shows the PicoClaw interface. On the left is a sidebar with icons for Overview, Channels, Models, Providers, Tools, Database, Skills (which is selected and highlighted in purple), and Logs. Below the sidebar is the version number v1.0.0. The main area has a header "Installed Capabilities" with a count of "7 Active". It lists seven skills: github, hardware, skill-creator, summarize, tmux, vision-reader, and weather, each with a brief description and an "Uninstall" button. Below this is a section titled "Skill Directory" with the sub-header "Official Picoclaw Repository". It contains the message "You have installed all available capabilities."

The extensibility hub for PicoClaw.

- - **Installed Skills**: Lists all skills currently loaded from `~/.picoclaw/workspace/skills` .
- - **Installation**: Install new skills directly from GitHub repositories.
- - **Management**: Uninstall skills that are no longer needed.

## 8. Settings

The screenshot shows the PicoClaw system settings interface. On the left is a sidebar with navigation links: Overview, Channels, Models, Providers, Tools, Database, Skills, and Settings (which is selected and highlighted in purple). Below the sidebar is the version number v1.0.0. The main content area is titled "System Settings" and contains two sections: "Agent Defaults" and "System Core".

**Agent Defaults**

- Primary Model: gemini-3-flash
- Max Tokens: 8192
- Restrict to Workspace:  Enable strict workspace boundaries

**System Core**

- Persistence Type: SQLite
- Gateway Port: 18790

In the top right corner of the main content area is a "Save Changes" button.

Advanced configuration portal.

- View and modify the `config.json` file directly from the UI.
- Update agent behavior, provider keys, and system thresholds.

## 9. Logs

The screenshot shows the PicoClaw application interface. On the left is a sidebar with navigation links: Overview, Channels, Models, Providers, Tools, Database, Skills, Settings, and Logs (which is highlighted). At the top right are buttons for 'Polling every 5s' and 'Auto-scroll: ON'. The main area is titled 'Live System Logs' and displays a scrollable list of log entries from a file named 'picoclaw.log'. The logs show interactions between a Telegram bot and an AI agent, including message processing, tool calls (like reading files), and responses to user queries about objects.

```

[2026-02-24T08:37:34Z] [INFO] [telegram] Starting Telegram bot (polling mode)...
[2026-02-24T08:37:34Z] [INFO] [channels] Outbound dispatcher started
[2026-02-24T08:37:35Z] [INFO] [telegram] Telegram bot connected {"username":"picoclaw79_bot"}
[2026-02-24T08:37:35Z] [INFO] [channels] All channels started
[2026-02-24T08:39:52Z] [INFO] [agent] Processing message from telegram:1851498538: {image: photo}
{"channel": "telegram", "chat_id": "1851498538", "sender_id": "1851498538", "session_key": ""}
[2026-02-24T08:39:52Z] [INFO] [agent] Routed message {"agent_id": "main", "matched_by": "default", "session_key": "agent:main:main"}
[2026-02-24T08:39:56Z] [INFO] [agent] LLM requested tool calls {"agent_id": "main", "count": 1, "iteration": 1, "tools": ["read_file"]}
[2026-02-24T08:39:56Z] [INFO] [agent] Tool call: read_file({"path": "/home/pi/picoclaw/workspace/skills/vision-reader/SKILL.md"})
{"agent_id": "main", "iteration": 1, "tool": "read_file"}
[2026-02-24T08:39:56Z] [INFO] [tool] Tool execution started {"args": {"path": "/home/pi/picoclaw/workspace/skills/vision-reader/SKILL.md"}, "tool": "read_file"}
[2026-02-24T08:39:56Z] [INFO] [tool] Tool execution completed {"duration_ms": 8, "result_length": 1825, "tool": "read_file"}
[2026-02-24T08:40:02Z] [INFO] [agent] LLM response without tool calls (direct answer) {"agent_id": "main", "content_chars": 559, "iteration": 2}
[2026-02-24T08:40:02Z] [INFO] [agent] Response: This is a photo of a **matte red water bottle** (it looks like a LARQ or similar self-cleaning style bottle)
sitting ... {"agent_id": "main", "final_length": 559, "iterations": 2, "session_key": "agent:main:main"}
[2026-02-24T08:40:02Z] [INFO] [agent] Processing message from telegram:1851498538: What is
this {"channel": "telegram", "chat_id": "1851498538", "sender_id": "1851498538", "session_key": ""}
[2026-02-24T08:40:02Z] [INFO] [agent] Routed message {"agent_id": "main", "matched_by": "default", "session_key": "agent:main:main"}
[2026-02-24T08:40:10Z] [INFO] [agent] LLM response without tool calls (direct answer) {"agent_id": "main", "content_chars": 698, "iteration": 1}
[2026-02-24T08:40:10Z] [INFO] [agent] Response: This is a **LARQ Bottle PureVis** (specifically in the "Canyon Red" or a similar deep red color). It's a self-
clean... {"agent_id": "main", "final_length": 698, "iterations": 1, "session_key": "agent:main:main"}
[2026-02-24T08:40:25Z] [INFO] [agent] Processing message from telegram:1851498538: What is this [image: photo]
{"channel": "telegram", "chat_id": "1851498538", "sender_id": "1851498538", "session_key": ""}
[2026-02-24T08:40:25Z] [INFO] [agent] Routed message {"agent_id": "main", "matched_by": "default", "session_key": "agent:main:main"}
[2026-02-24T08:40:30Z] [INFO] [agent] LLM response without tool calls (direct answer) {"agent_id": "main", "content_chars": 768, "iteration": 1}
[2026-02-24T08:40:30Z] [INFO] [agent] Response: This is an **iClever Vertical Ergonomic Mouse**. Specifically, it looks like the **iClever MD165** or a very
similar ... {"agent_id": "main", "final_length": 768, "iterations": 1, "session_key": "agent:main:main"}

```

A live view of the system logs.

- Displays the last 100 lines of the `picoclaw.log` file for debugging and monitoring.

## Backend API Reference

|  | Description                                    |  |
|--|--|--|
|  | Returns system metrics and metadata.           |  |
|  | Retrieves or updates the system configuration. |  |
|  | Returns the last 100 lines of system logs.     |  |
|  | Lists all installed skills.                    |  |
|  | Fetches available skills from the registry.    |  |
|  | Triggers a skill installation from GitHub.     |  |
|  | Uninstalls a specific skill.                   |  |
|  | Lists all database tables.                     |  |
|  | Retrieves rows from a specified table.         |  |
|  | Deletes a specific row by ID.                  |  |

|  | Description                            |  |
|--|--|--|
|  | Wipes an entire table.                 |  |
|  | Lists configured AI models.            |  |
|  | Returns channel status information.    |  |
|  | Lists configured LLM providers.        |  |
|  | Lists all registered tool definitions. |  |

## Technical Architecture

- - **Backend**: Built with Go, utilizing a modular architecture for providers, channels, and tools.
- - **Frontend**: A React SPA (Single Page Application) built with TypeScript and Vite, served via an embedded filesystem in the Go binary.
- - **Persistence**: SQLite for message history and agent state.
- - **Service Management**: Deployable via Systemd on Linux (Raspberry Pi).