

Pull Request: Native Tools, OpenAI Consolidation, and Stability Fixes

Summary

This release (v0.6.0) significantly enhances the Agent's capabilities by adding **Native Tools** for direct interaction with Home Assistant (Services, States, Templates, Logs). It also consolidates the logic to use the **OpenAI-compatible API only**, removing the complex and crash-prone Gemini native client.

Changes

New Native Tools (Capabilities)

The Agent can now "see" and "test" your Home Assistant instance directly:

| Tool Name | Description |
|--------------------------------------|--|
| <code>get_services(domain)</code> | Lists available services (e.g., <code>light.turn_on</code> , <code>script.reload</code>) to prevent hallucinated service calls. |
| <code>get_entity_states(id)</code> | Checks the current state/attributes of entities to verify if automations are triggering correctly. |
| <code>validate_template(tmp1)</code> | Renders Jinja2 templates (e.g., <code>{{ states('sensor.time') }}</code>) to verify logic before applying changes. |
| <code>check_config()</code> | Triggers a Home Assistant Core configuration check to ensure YAML validity. |
| <code>read_logs(lines)</code> | Reads the tail of <code>home-assistant.log</code> to diagnose errors after a failed reload. |

OpenAI Consolidation

- **Removed Native Gemini Client:** The project now uses the standard OpenAI-compatible client for *all* providers (Gemini, OpenAI, Claude, Local).
- **Simplified Dependency Chain:** Removed `google-genai` dependency to prevent conflicts and installation issues.
- **Unified Logic:** All AI interactions now flow through a single, stable code path.

Bug Fixes

- **Fixed "Bad Gateway" (502) Error:** Resolved a critical startup crash caused by an `IndentationError` in `agent_system.py`.
- **Fixed Crash Loop:** Removed orphaned code blocks that were causing the add-on to exit immediately.

Files Changed

| File | Description |
|--|---|
| <code>src/agents/tools.py</code> | Added implementation for 5 new tools (<code>get_services</code> , <code>read_logs</code> , etc.) |
| <code>src/ha/ha_websocket.py</code> | Added WebSocket support for service listing, state fetching, and template rendering |
| <code>src/agents/agent_system.py</code> | Registered new tools with OpenAI client; removed Gemini native code |
| <code>requirements.txt</code> | Cleaned up dependencies |
| <code>config.yaml</code> , <code>build.yaml</code> , <code>manifest.json</code> | Version bump to 0.6.0 |

Testing

- **Native Tools:** Verified that the agent can successfully list services and read logs via WebSocket/File API.
- **Stability:** Verified that the add-on starts successfully and maintains a stable connection.
- **OpenAI Streaming:** Verified that chat responses stream correctly without the native Gemini client.

📄 Documentation Upgrade

- **Service Schema Awareness:** Updated the Agent's system prompt to explicitly inform it that `get_services` returns full field schemas (arguments, types, selectors). This "unlocks" the ability for the agent to validate its own service calls against the actual definitions in your specific Home Assistant instance.

⚡ Stability Fix

- **History Sanitization:** Added automatic filtering of "orphaned" tool messages in the conversation history. This prevents the "Invalid parameter: messages with role 'tool'..." error (400 Bad Request) that could occur if the chat history lost synchronization with the backend.

Version: 0.6.3