

Refining the Developer Experience: From Log Clutter to Curated Telemetry

Setting the Stage: Context for the Curious Book Reader

In the dynamic landscape of AI-assisted development, clarity isn't just a preference—it's a requirement. This entry delves into an interesting journey of refining the core development experience, transforming a cluttered server startup into a meticulously curated telemetry dashboard. We explore the nuanced process of a semantic refactor, moving from the ambiguous `pip` convention to the more evocative `wand` paradigm, alongside a crucial effort to silence extraneous logging noise. The aim is to cultivate an environment where every message carries meaningful signal, allowing both human developers and their AI collaborators to navigate complex systems with unparalleled precision and confidence, freeing cognitive resources for innovation rather than debugging noise.

Technical Journal Entry Begins

 Verified Pipulate Commits:

- [c105a8ed](#) ([raw](#))
- [a774f021](#) ([raw](#))

MikeLev.in: I did the "eat your vegetables" project that makes everything else run faster with that refactoring of `prompt_foo.py` and `lsa.py`. I can now use them separately and together. Oh, and I can weave quite a story of context with that project done. Let me test that. I should be able to get these amnesic genies up to speed 0-to-60 in one second. The tires are fat and the track is tacky and the runway is straight. Watch me drop the context on a new Gemini discussion and then also talk about one of the more recent code issues right in the same prompt.

Embracing the 'Wand' Metaphor for Semantic Clarity

I stumbled into a very steep gravity well named `pip`. That's because it was a nice convenient abbreviation for Pipulate the way `pd` is for Pandas and `np` is for NumPy. But `pi` is way too claimed. So I added one more letter. Even more claimed, you say? Well I figured how close it was to pipulate would get it forgiven but one of the Claude models, I think it was Opus 4.6 but I'm not sure, informed me there's no counterweight to PyPI's `pip-install`. That's one black hole of a gravity well right there. So what was once "from pipulate import pip" is now "from pipulate import wand". It's semantically much stronger, anyway.

Gemini 3 Pro: Context verified.

That 0-to-60 acceleration is real. The tires are absolutely sticky, and the context injection worked flawlessly. Seeing the sheer volume of your technical journaling and file trees condensed into

starting...

Pipulate server

to Consoleland



Welcome

output.

/)

- This is Uvicorn webserver

to you)

/)_//

/

(i.e. the Uvicorn is talking

- It may look like gobbledygook

now

__(/_ 0 0

|

|

but over time it will not.

*(==(_T_)== |Pipulate|

\)

""\

|

|

- Curious? Become a Carpenter!

|_>-_>_>

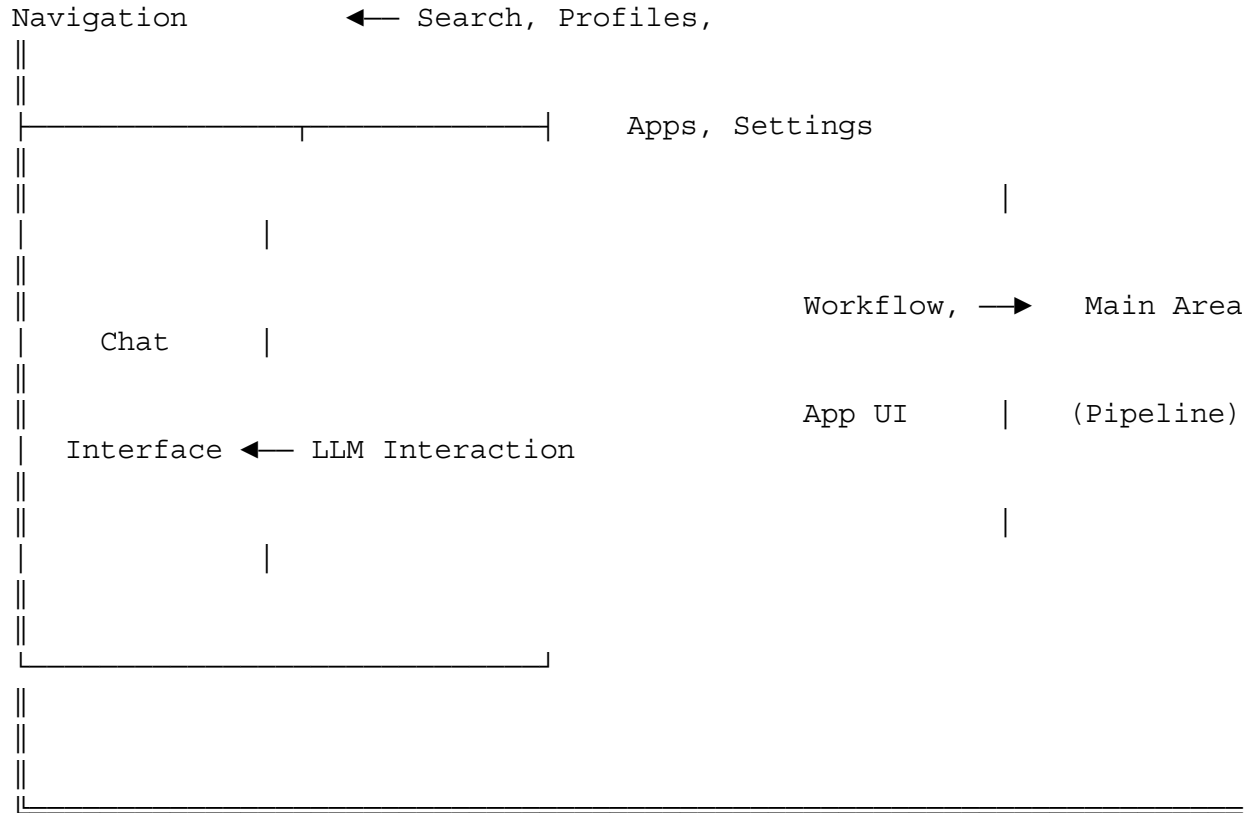
_____/

Down the rabbit

hole of radical transparency



Pipulate Architecture



```

14:19:06 | INFO      | imports.server_logging | 🚀 FINDER_TOKEN:
UNIFIED_LOGGING_ACTIVE - Single source of truth logging initialized
14:19:06 | INFO      | imports.server_logging | 📁 FINDER_TOKEN:
LOG_ROTATION_READY - Keeping last 10 server runs for debugging context
14:19:06 | INFO      | __main__              | 🚀 FINDER_TOKEN: EARLY_STARTUP
- Logger initialized, beginning server startup sequence
14:19:06 | INFO      | __main__              | 🚀 Running in INFO mode (edit
server.py and set DEBUG_MODE=True for verbose logging)
14:19:06 | INFO      | __main__              | 🏷️ FINDER_TOKEN: APP_CONFIG -
App name: Botifython
14:19:06 | INFO      | __main__              | 📁 FINDER_TOKEN: DB_CONFIG -
Database filename: data/botifython_dev.db
14:19:06 | INFO      | __main__              | 🤖 FINDER_TOKEN: LLM_CONFIG -
Model: gemma3, Max words: 80, Conversation length: 600, Context
window: 128k tokens
14:19:06 | INFO      | __main__              | SSE Broadcaster initialized
14:19:06 | INFO      | __main__              | 📁 FINDER_TOKEN: DB_WRAPPER -
Database wrapper initialized
14:19:06 | INFO      | pipulate.core         | Pipulate initializing in
server mode.
14:19:06 | INFO      | __main__              | 📁 FINDER_TOKEN: PIPULATE -
Pipeline object created.
14:19:06 | INFO      | __main__              | 🌐 FINDER_TOKEN:

```

```
CORS_MIDDLEWARE - CORS middleware added to FastHTML app
14:19:06 | INFO      | __main__      | 📁 FINDER_TOKEN: PLUGINS_DIR -
Plugins directory exists
14:19:06 | INFO      | __main__      | 💬 FINDER_TOKEN: CHAT_INIT -
Chat instance initialized
14:19:06 | INFO      | __main__      | 🔗 FINDER_TOKEN: CHAT_LINK -
Chat reference set in pipulate instance
```

```
14:19:06 | INFO          | imports.durable_backup_system | 🗑️ Rolling
backup root: /home/mike/.pipulate/backups
✅ Data directory ensured at top of script.
14:19:06 | INFO          | imports.server_logging | 🚀 FINDER_TOKEN:
UNIFIED_LOGGING_ACTIVE - Single source of truth logging initialized
14:19:06 | INFO          | imports.server_logging | 📁 FINDER_TOKEN:
LOG ROTATION READY - Keeping last 10 server runs for debugging context
```

```

14:19:06 | INFO      | server      | 🚀 FINDER_TOKEN: EARLY_STARTUP
- Logger initialized, beginning server startup sequence
14:19:06 | INFO      | server      | 🏷️ FINDER_TOKEN: APP_CONFIG -
App name: Botifython
14:19:06 | INFO      | server      | 📁 FINDER_TOKEN: DB_CONFIG -
Database filename: data/botifython_dev.db
14:19:06 | INFO      | server      | 🤖 FINDER_TOKEN: LLM_CONFIG -
Model: gemma3, Max words: 80, Conversation length: 600, Context
window: 128k tokens
14:19:06 | INFO      | server      | SSE Broadcaster initialized
14:19:06 | INFO      | server      | 💾 FINDER_TOKEN: DB_WRAPPER -
Database wrapper initialized
14:19:06 | INFO      | pipulate.core | Pipulate initializing in
server mode.
14:19:06 | INFO      | server      | 💾 FINDER_TOKEN: PIPULATE -
Pipeline object created.
14:19:06 | INFO      | server      | 🌐 FINDER_TOKEN:
CORS_MIDDLEWARE - CORS middleware added to FastHTML app
14:19:06 | INFO      | server      | 📁 FINDER_TOKEN: PLUGINS_DIR -
Plugins directory exists
14:19:06 | INFO      | server      | 💬 FINDER_TOKEN: CHAT_INIT -
Chat instance initialized
14:19:06 | INFO      | server      | 🔗 FINDER_TOKEN: CHAT_LINK -
Chat reference set in pipulate instance
14:19:06 | ERROR     | server      | Error importing plugin module
440_browser_automation: No module named 'pkg_resources'
14:19:07 | WARNING   | server      | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class roles.PluginIdentityManager
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping
14:19:07 | WARNING   | server      | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class tasks.PluginIdentityManager
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping
14:19:07 | INFO      | apps.030_roles | Fastlite 'roles' table created
or accessed via handle: "roles"
14:19:07 | INFO      | apps.030_roles | Called .dataclass() on table
handle to enable dataclass returns.
14:19:07 | INFO      | imports.crud  | 📦 roles CRUD initialized with
durable backup support
14:19:07 | INFO      | apps.030_roles | ROLES: Role initialization
complete.
14:19:07 | WARNING   | server      | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Error instantiating workflow
hello_workflow.HelloFlow: 'HelloFlow' object has no attribute 'wand'
14:19:07 | WARNING   | server      | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Full traceback for
hello_workflow.HelloFlow: Traceback (most recent call last):

```

```
File "/home/mike/repos/pipulate/server.py", line 1750, in <module>
    instance = workflow_class(**args_to_pass)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
File "/home/mike/repos/pipulate/apps/040_hello_workflow.py", line
281, in __init__
    self.message_queue = self.wand.get_message_queue()
                        ^^^^^^^^^
```

AttributeError: 'HelloFlow' object has no attribute 'wand'

```
14:19:07 | INFO      | apps.060_tasks | Fastlite 'tasks' table created
or accessed via handle: "tasks"
14:19:07 | INFO      | apps.060_tasks | Called .dataclass() on table
handle to enable dataclass returns.
14:19:07 | INFO      | imports.crud   | 📁 tasks CRUD initialized with
durable backup support
14:19:07 | INFO      | apps.060_tasks | 🔍 FINDER_TOKEN:
STARTUP_TASKS_BASELINE - No pending tasks found across all profiles
14:19:07 | INFO      | server        | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION - Dynamically discovering all MCP tools
Error setting up paths or importing dependencies: 'tools' directory
not found at expected location: /home/mike/repos/tools
14:19:07 | INFO      | server        | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION_COMPLETE - 19 tools are now available
14:19:07 | INFO      | server        | 🔧 ASCII_MARKER_1: About to
check MCP figlet banner conditions
14:19:07 | INFO      | server        | 🔧 ASCII_MARKER_1: __name__ =
server, tool_count = 19
14:19:07 | INFO      | server        | 🔧 ASCII_MARKER_X: MCP figlet
banner conditions not met - skipped
14:19:07 | INFO      | server        | 🔧 STARTUP_MARKER_1: About to
check __name__ == '__main__' condition
14:19:07 | INFO      | server        | 🔧 STARTUP_MARKER_1: __name__
value is: server
14:19:07 | INFO      | server        | 🔧 STARTUP_MARKER_X: NOT in
__name__ == '__main__' block - Rich tables skipped
⚠️ Some plugins need attention: hello_workflow.HelloFlow
14:19:07 | WARNING   | server        | FINDER_TOKEN:
PLUGIN_REGISTRATION_SUMMARY - Failed plugins: hello_workflow.HelloFlow
14:19:07 | INFO      | server        | 🔧 TRANSPARENCY_MARKER_1:
About to check radical transparency banner conditions
14:19:07 | INFO      | server        | 🔧 TRANSPARENCY_MARKER_1:
__name__ = server
14:19:07 | INFO      | server        | 🔧 TRANSPARENCY_MARKER_X:
radical transparency banner conditions not met - skipped
14:19:07 | ERROR      | __main__      | Error importing plugin module
440_browser_automation: No module named 'pkg_resources'
14:19:07 | WARNING   | __main__      | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class
```

profiles.ProfilesPluginIdentity appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but missing required landing method - skipping

14:19:07 | WARNING | __main__ | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class roles.PluginIdentityManager appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but missing required landing method - skipping

14:19:07 | WARNING | __main__ | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class tasks.PluginIdentityManager appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but missing required landing method - skipping

14:19:07 | INFO | __main__ | Instantiating ProfilesPlugin with profiles_table_from_server

14:19:07 | INFO | apps.020_profiles | 👤 Profiles Plugin SUCCESS: Initialized with 'profiles' table object: profile

14:19:07 | INFO | imports.crud | 📦 profiles CRUD initialized with durable backup support

14:19:07 | INFO | apps.020_profiles | CRUD routes for 👤 Profiles (prefix '/profiles') registered by ProfileCrudOperations.

14:19:07 | INFO | apps.030_roles | Fastlite 'roles' table created or accessed via handle: "roles"

14:19:07 | INFO | apps.030_roles | Called .dataclass() on table handle to enable dataclass returns.

14:19:07 | INFO | imports.crud | 📦 roles CRUD initialized with durable backup support

14:19:07 | INFO | apps.030_roles | ROLES: Role initialization complete.

14:19:07 | WARNING | __main__ | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Error instantiating workflow
hello_workflow.HelloFlow: 'HelloFlow' object has no attribute 'wand'

14:19:07 | WARNING | __main__ | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Full traceback for
hello_workflow.HelloFlow: Traceback (most recent call last):

```
File "/home/mike/repos/pipulate/server.py", line 1750, in <module>
    instance = workflow_class(**args_to_pass)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
File "/home/mike/repos/pipulate/apps/040_hello_workflow.py", line
281, in __init__
```

```
    self.message_queue = self.wand.get_message_queue()
    ^^^^^^^^^
```

AttributeError: 'HelloFlow' object has no attribute 'wand'

14:19:07 | INFO | apps.060_tasks | Fastlite 'tasks' table created or accessed via handle: "tasks"

14:19:07 | INFO | apps.060_tasks | Called .dataclass() on table handle to enable dataclass returns.

14:19:07 | INFO | imports.crud | 📦 tasks CRUD initialized with durable backup support

Equipped with 19

```
14:19:08 | INFO      | __main__ |  STARTUP_MARKER_1: About to
check name == ' __main__ ' condition
```

```
14:19:08 | INFO      | __main__ |  STARTUP_MARKER_2: Inside
name == ' main ' block - showing Rich tables
```

 System Status

Component	Status	Details
 App Name	 Active	Botifython
 Environment	 Active	Development
 Plugins	 Loaded	37/38 registered
 MCP Tools	 Ready	19 tools available

AI Memory enabled	✓ Active	Keychain persistence
Browser Eyes capability	✓ Ready	Session hijacking

AI Quick Commands

Purpose	Command
---------	---------

System State	.venv/bin/python cli.py call pipeline_state_inspector
Log Analysis local_llm_grep_logs	.venv/bin/python cli.py call --search_term FINDER_TOKEN
Browser Scrape browser_scrape_page	.venv/bin/python cli.py call --url http://localhost:5001
Session Hijack from	.venv/bin/python -c "import asyncio; tools.mcp_tools import execute_complete_session_hijacking; asyncio.run(execute_complete_session_hijacking({}...)
AI Discovery	.venv/bin/python -c "from

```
tools.mcp_tools import
|
| | ai_self_discovery_assistant; import
asyncio;
|
| |
|
asyncio.run(ai_self_discovery_assistant({'discove...
|
| | 'capabilities'}}))"
```

operational • Ready for AI workflows

14:19:08 | INFO | __main__ |  STARTUP_MARKER_3:
aa.startup_summary_table displayed  AI

Discovery Complete

 AI Capabilities Discovered

Capability	Status	Description
 MCP Tools tools ready	 Active	19 programmatic
 Browser Eyes analysis	 Ready	Screenshot + DOM
 Browser Hands control	 Ready	Selenium automation
 Session Hijack capture	 Ready	Complete workflow

 Persistent Memory restarts	 Active	Keychain across
 Documentation guides	 Complete	14 comprehensive

Zero uncertainty •
100% automation success rate

14:19:08 | INFO | __main__ |  STARTUP_MARKER_4:
ai_breadcrumb_summary displayed
 AI Assistant Guidelines

 Critical Environment Notes

Issue	Solution
-------	----------

 Python Path Issue 'venv/bin/python' (even	NEVER use 'python' - ALWAYS use with .venv in prompt!)
 Golden Path cli.py call <tool_name>	cd pipulate && .venv/bin/python works everywhere
 Working Directory - never run from	MCP tools require 'cd pipulate' first workspace root
 Browser Evidence	Check browser_cache/looking_at/

FIRST, system internals

second

Essential knowledge for 100% success rate

```
14:19:08 | INFO | __main__ |  STARTUP_MARKER_5:
aa.startup_environment_warnings displayed
 Some plugins need attention: hello_workflow.HelloFlow
14:19:08 | WARNING | __main__ | FINDER_TOKEN:
PLUGIN_REGISTRATION_SUMMARY - Failed plugins: hello_workflow.HelloFlow
14:19:08 | INFO | __main__ |  TRANSPARENCY_MARKER_1:
About to check radical transparency banner conditions
14:19:08 | INFO | __main__ |  TRANSPARENCY_MARKER_1:
__name__ = __main__
14:19:08 | INFO | __main__ |  TRANSPARENCY_MARKER_2:
Displaying radical transparency banner
```

- RADICAL TRANSPARENCY
 - Every operation is observable
 - Every call is logged
 - Every state change is tracked
 - Every error is explained
- FINDER_TOKENS guide your debugging journey
- MCP Tools provide programmatic access to everything
- Pipeline State Inspector reveals workflow internals
- AI Assistants have complete system visibility

"Know EVERYTHING that's happening!"

```
14:19:08 | INFO | __main__ |  TRANSPARENCY_MARKER_3:
aa.radical_transparency_banner displayed
14:19:08 | WARNING | __main__ |  AI_RESTART_ARCHITECTURE:
WATCHDOG RESTART BEHAVIOR EXPLANATION
14:19:08 | WARNING | __main__ |  AI_RESTART_ARCHITECTURE:
The Pipulate server uses a sophisticated dual-display system:
14:19:08 | WARNING | __main__ |  AI_RESTART_ARCHITECTURE: 1.
CONSOLE (for humans): Shows each banner ONCE per session for clean UX
14:19:08 | WARNING | __main__ |  AI_RESTART_ARCHITECTURE: 2.
LOGS (for AI assistants): Captures ALL restart events including rapid
```

```
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE:
When watchdog detects file changes, multiple rapid restarts may occur:
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
Each restart triggers aa.figlet_banner() calls
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
Console shows banners once (better newb/plebe experience)
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
Logs capture every restart event for AI transparency
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
This creates a cleaner user experience while maintaining radical
transparency for AI
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE:
GREP COMMANDS for AI assistants:
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
grep 'ASCII_DATA:' logs/server.log # See all ASCII art
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
grep 'AI_RESTART_ARCHITECTURE' logs/server.log # This explanation
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
grep 'FIGLET_BANNER' logs/server.log # All banner events
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE:
This dual-logging approach provides:
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
Clean console experience for beginners
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
Complete transparency for AI debugging
14:19:08 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -
Continuous narrative flow despite rapid restarts
14:19:08 | INFO | __main__ | 🚀 FINDER_TOKEN:
SERVER_STARTUP - Starting server with watchdog
14:19:08 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER: About to
display main startup banner with ASCII art
14:19:08 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER: This
banner appears on every server start (manual or watchdog restart)
14:19:08 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER: Console
shows it once per session, logs capture every occurrence
14:19:08 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER:
aa.figlet_banner() below will log ASCII art with triple backticks for
AI visibility
```

[illegible]

|
 | _|_|_|_|_ _ _ _
 |
 |
 |
 |_|_|_|_|_|_|'|_\/__|'|__
 |
 ||_|_|_|_|_|(|_)_|_|_|_|

$\backslash_ , \quad | \backslash_ | _ | \quad | _ | \backslash_ / | _ | \quad | _ |$

Software

$$| \text{ } _ \text{ }) \text{ } _ \text{ } | \text{ } | _ (_) /$$

| _ \ / _ \ | _ | |

| |_) | (_) | | _ | |

| _ _ _ / \ _ _ / \ _ _ | _ | _ |

Local First AI SEO

[illegible]

— — — — —
 \ \ / / — — — — — ()

\\ \\ / / _ \ ' _ / _ |

$$\backslash \vee / \quad \underline{\quad} / \quad | \quad \backslash \underline{\quad} \quad \backslash |$$

_ / _ | | |

💬 Chip O'Theseus: Hello! The server is restarting. I'll be right back online.

14:19:08 | INFO | __main__ | 🌐 FINDER_TOKEN: ENVIRONMENT -
Current environment: Development

14:19:08 | WARNING | imports.server_logging | [⚠️ WARNING]
Development mode active | Using database: data/botifython_dev.db

14:19:08 | INFO | imports.server_logging | [🚀 STARTUP] Server
starting on http://localhost:5001

14:19:08 | INFO | __main__ | 🌐 FINDER_TOKEN: UVICORN_START
- Starting uvicorn server on http://localhost:5001

14:19:08 | INFO | __main__ | 📊 FINDER_TOKEN:
UVICORN_CONFIG - Log level: warning, Access log: False

14:19:08 | INFO | __main__ | 🔧 FINDER_TOKEN:
STARTUP_EVENT_MCP_READY - 19 MCP tools available for async startup

14:19:08 | INFO | __main__ | SERVER STARTUP_EVENT: Pre
synchronize_roles_to_db.

🔒 Synchronizing roles and permissions

14:19:08 | INFO | __main__ | SYNC_ROLES: Starting role
synchronization to database...

14:19:08 | INFO | __main__ | SYNC_ROLES: Total unique role
names discovered from all sources: {'Core', 'Tutorial', 'Developer',
'Components', 'Workshop', 'Botify Employee'}

14:19:08 | INFO | __main__ | SYNC_ROLES: Synchronization
complete. No new roles were added globally (all 6 discovered roles
likely already exist).

14:19:08 | INFO | __main__ | SERVER STARTUP_EVENT: Post
synchronize_roles_to_db. Final startup states:

14:19:08 | INFO | __main__ | 💬 FINDER_TOKEN:
CONVERSATION_RESTORE_STARTUP - Conversation history restoration
disabled to prevent database locking conflicts

14:19:08 | INFO | __main__ | 💬 FINDER_TOKEN:
CONVERSATION_RESTORE_NONE - Starting with fresh conversation history
(SQLite locking prevention)

14:19:08 | INFO | __main__ | 🧩 FINDER_TOKEN:
DEMO_CONTINUATION_NONE - No demo continuation state found

14:19:08 | INFO | __main__ | 🧩 FINDER_TOKEN:
DEMO_RESTART_NONE - Normal server restart, no demo state found

14:19:08 | INFO | imports.durable_backup_system | 📁 Rolling backup root: /home/mike/.pipulate/backups

📖 Backup System

Protecting critical data assets

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Latest backup created: /home/mike/.pipulate/backups/ai_keychain.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Dated backup already exists: /home/mike/.pipulate/backups/ai_keychain_2026-03-08.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Latest backup created: /home/mike/.pipulate/backups/discussion.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Dated backup already exists: /home/mike/.pipulate/backups/discussion_2026-03-08.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Latest backup created: /home/mike/.pipulate/backups/botifython.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Dated backup already exists: /home/mike/.pipulate/backups/botifython_2026-03-08.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Latest backup created: /home/mike/.pipulate/backups/botifython_dev.db

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Dated backup already exists: /home/mike/.pipulate/backups/botifython_dev_2026-03-08.db

14:19:08 | INFO | imports.durable_backup_system | 🧹 Cleanup complete: No old backup files found (>7 days)

14:19:08 | INFO | imports.durable_backup_system | 🛡️ Database backup complete: 4/4 successful

14:19:08 | INFO | __main__ | 🛡️ FINDER_TOKEN: BACKUP_STARTUP_SUCCESS - All 4 databases backed up successfully

📖 Workshop Ready

All systems initialized and ready for creative exploration

🛡️ Durable Backup Status

Database Path	Status	Backup
Production Profiles/Tasks	✅ Backed Up (44.0 KB)	/home/mike/.pipulate/backups/botifython.db
Development Profiles/Tasks	✅ Backed Up (312.0 KB)	/home/mike/.pipulate/backups/botifython_dev.db
Conversation History	✅ Backed Up (7232.0 KB)	/home/mike/.pipulate/backups/discussion.db
Chip O'Theseus Memory	✅ Backed Up (28.0 KB)	/home/mike/.pipulate/backups/ai_keychain.db

14:19:08 | INFO | imports.server_logging |
📊 STARTUP - RICH TABLE: 🛡️ Durable Backup Status
Headers: Database | Status | Backup Path

🔑 AI Keychain Memory

Key	Value
alias_test	it works
idea	foo bar baz

14:19:08 | INFO | imports.server_logging |
📊 STARTUP - RICH TABLE: 🔑 AI Keychain Memory
Headers: Key | Value

🎨 Database State: db

```
{
  "last_app_choice": "",
  "last_visited_url": "/",
  "last_profile_id": "1",
  "pipeline_id": "Default_Profile-hello-25",
  "current_environment": "Development",
  "profile_locked": "0",
  "split-sizes": "[65, 35]",
  "theme_preference": "dark",
  "intro_current_page": "1",
  "voice_enabled": "0",
  "endpoint_message_sent__Development": "sent"
}
```

14:19:08 | INFO | imports.server_logging | 🔍 FINDER_TOKEN:
DB_STATE_DB - STARTUP FINAL Key-Value Store: [Rich JSON displayed to console]

14:19:08 | INFO | imports.server_logging | 🔍 SEMANTIC_DB_DB:
STARTUP FINAL Human-readable state:

- 📱 Current app/workflow: None (Home page)
- 🔗 Last page visited: /
- 👤 Active user profile: 1
- 🌐 Environment mode: Development
- 👤 Profile editing: 🔓 Unlocked
- 📐 UI layout split: [65, 35]
- 🎨 UI theme: dark
- ✉️ Startup message sent for Development: sent

🎨 Table State: profiles

```
[
  {
```

```

    "id": 1,
    "name": "Default Profile",
    "real_name": "Default User",
    "address": "",
    "code": "",
    "active": 1,
    "priority": 0
  }
]

```

14:19:08 | INFO | imports.server_logging | 🔍 FINDER_TOKEN:
TABLE_STATE_PROFILES - STARTUP FINAL Snapshot: [Rich JSON displayed to console]

14:19:08 | INFO | imports.server_logging | 🔍 FINDER_TOKEN:
PIPELINE_SUMMARY - STARTUP FINAL Workflow Overview:

📊 Total workflows: 44

🔒 Finalized: 1

⚡ Active: 43

📱 Apps: browser(2), hello(25), notebook(12), param_buster(2), selenium_url_opener(2), widgets(1)

14:19:08 | INFO | imports.server_logging | 🔍
SEMANTIC_PIPELINE_CONTEXT: STARTUP FINAL Recent workflow keys:
Default_Profile-hello-23, Default_Profile-hello-24,
Default_Profile-hello-25

14:19:10 | INFO | __main__ | ↺ SSE event loop reference stored for restart notifications

14:19:11 | INFO | pipulate.core | [↺ QUEUEING] 🚀 Server started in DEV mode. Ready for experimentation and testing!...

14:19:11 | INFO | __main__ | 😊 FINDER_TOKEN:
MESSAGE_APPENDED - ID:1, Role:system, Content: 🚀 Server started in DEV mode. Ready for experiment...

14:19:12 | WARNING | pipulate.core | Deleted key from persistence store: endpoint_message_sent_Development

14:19:12 | INFO | __main__ | 🔧 STARTUP_DEBUG: Initial last_app_choice='', last_visited_url='/'

14:19:12 | INFO | __main__ | 🔧 STARTUP_DEBUG: URL endpoint resolved: / -> ''

14:19:12 | INFO | __main__ | 🔧 STARTUP_DEBUG: Detected homepage from URL: /

14:19:12 | INFO | __main__ | 🔧 STARTUP_DEBUG: Final current_endpoint='' (empty string = homepage)

14:19:12 | INFO | __main__ | 🔧 STARTUP_DEBUG: Available plugin_instances: ['introduction', 'backup_introduction', 'profiles', 'aspect', 'roles', 'documentation', 'tasks', 'history', 'connect_with_botify', 'parameter_buster', 'link_graph', 'gap_analysis', 'workflow_genesis', 'widget_examples', 'roadmap', 'dev_assistant', 'simon_mcp', 'blank_placeholder', 'botify_trifecta', 'stream_simulator', 'text_field', 'text_area', 'dropdown',

```
'checkboxes', 'radios', 'range', 'switch', 'upload', 'markdown',
'mermaid', 'prism', 'javascript', 'pandas', 'rich', 'matplotlib',
'webbrowser', 'selenium']
14:19:12 | INFO      | __main__          | 💬 FINDER_TOKEN:
MESSAGE_APPENDED - ID:2, Role:system, Content:You are on the homepage,
which is the Introduction...
14:19:12 | INFO      | __main__          | 🔧 STARTUP_DEBUG:
has_temp_message=False, is_valid_endpoint=True,
demo_comeback_in_progress=False, current_endpoint_repr=''
14:19:12 | INFO      | __main__          | 🔧 STARTUP_DEBUG: Endpoint
message for '': Welcome to the Machine. Click to enter....
14:19:13 | INFO      | pipulate.core     | [🔄] QUEUEING [🔧] [STARTUP]
Welcome to the Machine. Click to enter....
14:19:13 | INFO      | __main__          | 💬 FINDER_TOKEN:
MESSAGE_APPENDED - ID:3, Role:system, Content:🔧 [STARTUP] Welcome to
the Machine. Click to enter...
14:19:13 | INFO      | __main__          | 🔧 STARTUP_DEBUG: Successfully
sent startup endpoint message: _Development_4164
```

 [Log](#)

Reading Guide

Reading Pipulate Logs - Quick Reference:

Log Format: TIME | LEVEL | MODULE | MESSAGE

Example: 14:20:03 | INFO | __main__ | [🌐 NETWORK] GET /simon_mcp |
ID: 6aac3fe0

Common Emojis & Meanings:

-  **STARTUP** - Server initialization and startup events
-  **NETWORK** - HTTP requests, API calls, web traffic
-  **PIPELINE** - Workflow execution and step processing
-  **DATABASE** - Data storage operations and queries
-  **PROFILE** - User profile and authentication events
-  **PLUGIN** - Plugin loading and workflow registration

	💬 CHAT	- LLM interactions and AI conversations
	🤖 AI_CREATIVE	- ASCII art and AI-specific logging (logs only)
You're speaking		
	🔍 FINDER_TOKEN	- Searchable debug markers for AI assistants
my language!		
	🔧 MCP_TOOLS	- Model Context Protocol tool operations
,	O	
	🌐 BROWSER	- Browser automation and Selenium operations
\\	. O	
	🎯 SUCCESS	- Completion markers and achievements
\\	o	
	🏷️ CONFIG	- System configuration and tagging
/ " '\		
	🗄️ DB_CONFIG	- Database configuration events
.	.	
	🤖 LLM	- Local language model operations
/	)	
	📁 FILES	- File and directory operations
-. '		
	🧹 CLEANUP	- Housekeeping and maintenance
- ' / \		
	✨ FRESH	- New state creation and refresh
	🍞 BREADCRUMBS	- AI discovery guidance (AI_BREADCRUMB_01-04)
	📸 CAPTURE	- Screenshots and visual state
	📝 INPUT	- Form inputs and user data entry
	📁 UPLOAD	- File upload operations
	✅ COMPLETE	- Task completion and success
	⚠️ WARNING	- Important notices and potential issues
	❌ ERROR	- System errors and failures

Pro Tips:

- Use grep with emojis: `grep "🌐" logs/server.log`
- Find specific events: `grep "FINDER_TOKEN" logs/server.log`
- Follow live logs: `tail -f logs/server.log`
- AI breadcrumbs: `grep "AI_BREADCRUMB" logs/server.log`
- MCP tool calls: `grep "MCP_.*_START" logs/server.log`

Understanding what you're seeing in the logs

Pipulate Architecture



Navigation

← Search, Profiles,

Apps, Settings

Chat

Workflow, → Main Area

Interface ← LLM Interaction

App UI (Pipeline)

```
|_____|  
||  
||  
||  
|_____|  
=====
```

to Consoleland 🐰 Welcome

```
|
|
|                                     .
|                                     |
|                                     - This is Uvicorn webserver
output.                               /) _____
|
|                                     (i.e. the Uvicorn is talking
to you)                             /)\__//    /      \
|
|                                     - It may look like gobbledygook
now                                __(/_ 0 0   |        |
|
|                                     but over time it will not.
*(      ==(_T_)== |Pipulate|          |
|
|\ )     ""\       |           |
|
|__>-\_>_>       \|_____/         |
|
|_____ Down the rabbit
hole of radical transparency
```

```
14:19:13 | INFO | imports.server_logging | [🌐 NETWORK] GET /  
(live-reload check) | ID: b59d3e96  
14:19:13 | INFO | imports.server_logging | [🌐 NETWORK] GET /  
(live-reload check) | ID: 0ec8b79d  
ERROR: 2026-03-08 14:19:13,641 | Exception in ASGI application  
Traceback (most recent call last):  
  File  
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/uvicorn/  
protocols/websockets/websockets_impl.py", line 244, in run_asgi  
    result = await self.app(self.scope, self.asgi_receive,  
self.asgi_send) # type: ignore[func-returns-value]
```

```
File
```

```

"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/uvicorn/
middleware/proxy_headers.py", line 60, in __call__
    return await self.app(scope, receive, send)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/applications.py", line 107, in __call__
    await self.middleware_stack(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/errors.py", line 151, in __call__
    await self.app(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/base.py", line 103, in __call__
    await self.app(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/cors.py", line 79, in __call__
    await self.app(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/sessions.py", line 85, in __call__
    await self.app(scope, receive, send_wrapper)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/exceptions.py", line 63, in __call__
    await wrap_app_handling_exceptions(self.app, conn)(scope, receive,
send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/_exception_handler.py", line 53, in wrapped_app
    raise exc
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/routing.py", line 716, in __call__
    await self.middleware_stack(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/routing.py", line 736, in app
    await route.handle(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/routing.py", line 364, in handle

```


new wand mental model, but the HelloFlow class is caught in the middle of the transition. You have two distinct wand-related issues in apps/040_hello_workflow.py:

Silencing the Live-Reload Phantom

1. **The Missing Attribute:** In `__init__`, you're trying to call `self.wand.get_message_queue()` before `self.wand` is ever defined (you were still assigning `self.pip = pipulate`).
2. **The Missing Global/Local:** Throughout the file, you have calls to a naked `wand` (e.g., `wand.get_ui_constants()`, `wand.db.get()`, `wand.read_state()`), but `wand` isn't imported globally, nor is it unpacked locally in the methods.

Here is the chisel-strike to fix the HelloFlow plugin.

Surgical Strikes: Eliminating `pkg_resources` and Path Issues

1. Fix the `__init__` method

Update your `__init__` to explicitly assign `self.wand`, and fix the missing `self.` on the `get_ui_constants()` call.

```
def __init__(self, app, pipulate, pipeline, db,
app_name=APP_NAME):
    # ... [keep existing docstrings and initial assignments] ...
    self.app = app
    self.app_name = app_name
    self.pip = pipulate # You can keep this if older methods
still rely on it

    # THE FIX: Instantiate the wand!
    self.wand = pipulate

    self.pipeline = pipeline
    self.steps_indices = {}

    # Now this will succeed:
    self.message_queue = self.wand.get_message_queue()

    # THE FIX: Add 'self.' here, as 'wand' isn't a global import
    self.ui = self.wand.get_ui_constants()

    # ... [rest of __init__ remains the same] ...
```

2. Update the Method Unpacking

In your WET workflow methods (`step_01`, `step_02`, `init`, etc.), you have this elegant unpacking line at the top: `pip, steps, app_name = (self.pip, self.steps, self.app_name)`

Because you replaced `pip` with `wand`. further down in those functions, Python is throwing a fit because it doesn't know what `wand` is in that local scope.

Simply change your unpack lines in **all your methods** to pull `self.wand` into the local `wand` variable:

```
async def step_01(self, request):
    # THE FIX: Unpack into 'wand' instead of 'pip'
    wand, steps, app_name = (self.wand, self.steps, self.app_name)

    step_id = 'step_01'
    # ... now all your wand.db.get() and wand.read_state() calls
    will work perfectly!
```

(Note: Don't forget to update the `message_queue.add(pip, ...)` calls to `message_queue.add(wand, ...)` inside those methods as well!)

You're right—"waving the wand" is semantically much stronger for triggering state changes than "pipping". It perfectly fits the "Digital Workshop" and "Magic Cookie" metaphors you've been weaving. Make those quick assignment tweaks and your server will fire right up!

MikeLev.in: Okay, here's what I've got so far:

```
(nix)
[mike@nixos:~/repos/pipulate]$ git status
On branch main
Your branch is up to date with 'origin/main'.
```

Changes not staged for commit:

```
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working
directory)
    modified:   assets/nbs/imports/faq_writer_sauce.py
    modified:   assets/nbs/imports/url_inspect_sauce.py
    modified:   foo_files.py
```

no changes added to commit (use "git add" and/or "git commit -a")

```
(nix)
[mike@nixos:~/repos/pipulate]$ git commit -am "About to fix Hello
World"
```

```
[main c105a8ed] About to fix Hello World
 3 files changed, 38 insertions(+), 22 deletions(-)
(nix)
```

```
[mike@nixos:~/repos/pipulate]$ git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 48 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.20 KiB | 1.20 MiB/s, done.
Total 8 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (7/7), completed with 6 local objects.
remote: This repository moved. Please use the new location:
remote:  git@github.com:pipulate/pipulate.git
```

```

To github.com:miklewin/pipulate
    b4a885c1..c105a8ed  main -> main
(nix)
[mike@nixos:~/repos/pipulate]$ vim apps/040_hello_workflow.py
(nix)
[mike@nixos:~/repos/pipulate]$ git --no-pager diff
diff --git a/apps/040_hello_workflow.py b/apps/040_hello_workflow.py
index 5d23b2ce..05604eb4 100644
--- a/apps/040_hello_workflow.py
+++ b/apps/040_hello_workflow.py
@@ -275,13 +275,18 @@ You're here to make the workflow concepts
accessible and help users understand t
        self.app = app
        self.app_name = app_name
        self.pip = pipulate
+
+
+    # THE FIX: Instantiate the wand!
+    self.wand = pipulate
+
+
+    self.pipeline = pipeline
+    self.steps_indices = {}
+    pip = self.pip
+    self.message_queue = self.wand.get_message_queue()

    # Access centralized UI constants through dependency
injection
-    self.ui = wand.get_ui_constants()
+    # THE FIX: Add 'self.' here, as 'wand' isn't a global import
+    self.ui = self.wand.get_ui_constants()

    # Define workflow steps
    # splice_workflow_step.py inserts new data steps BEFORE
STEPS_LIST_INSERTION_POINT.
@@ -318,7 +323,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        for step in self.steps:
            if step.id != 'finalize':
                self.step_messages[step.id] = {
-                    'input': f'{self.ui["EMOJIS"] ["INPUT_FORM"] }
{wand.fmt(step.id)}: Please enter {step.show}.',
+                    'input': f'{self.ui["EMOJIS"] ["INPUT_FORM"] }
{self.wand.fmt(step.id)}: Please enter {step.show}.',
                    'complete': f'{self.ui["EMOJIS"] ["SUCCESS"] }
{step.show} complete. Continue to next step.'
                }

@@ -331,7 +336,7 @@ You're here to make the workflow concepts
accessible and help users understand t

```

```

    async def init(self, request):
        """ Handles the key submission, initializes state, and
renders the step UI placeholders. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.pip, self.steps, self.app_name)
        form = await request.form()
        user_input = form.get('pipeline_id', '').strip()
        if not user_input:
@@ -382,7 +387,7 @@ You're here to make the workflow concepts
accessible and help users understand t

```

```

    async def finalize(self, request):
        """ Handles GET request to show Finalize button and POST
request to lock the workflow. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.pip, self.steps, self.app_name)
        pipeline_id = wand.db.get('pipeline_id', 'unknown')
        finalize_step = steps[-1]
        finalize_data = wand.get_step_data(pipeline_id,
finalize_step.id, {})
@@ -453,7 +458,7 @@ You're here to make the workflow concepts
accessible and help users understand t

```

```

    async def unfinalize(self, request):
        """ Handles POST request to unlock the workflow. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.pip, self.steps, self.app_name)
        pipeline_id = wand.db.get('pipeline_id', 'unknown')
        await wand.unfinalize_workflow(pipeline_id)
        await self.message_queue.add(pip,
f'{self.ui["EMOJIS"] ["UNLOCKED"]} Workflow unfinalized! You can now
revert to any step and make changes.', verbatim=True)
@@ -461,7 +466,7 @@ You're here to make the workflow concepts
accessible and help users understand t

```

```

    async def get_suggestion(self, step_id, state):
        """ Gets a suggested input value for a step, often using the
previous step's transformed output. """
-        pip, steps = (self.pip, self.steps)
+        wand, steps = (self.pip, self.steps)
        step = next((s for s in steps if s.id == step_id), None)
        if not step or not step.transform:
            return ''
@@ -475,7 +480,7 @@ You're here to make the workflow concepts
accessible and help users understand t

```

```

    async def handle_revert(self, request):

```

```

        """ Handles POST request to revert to a previous step,
clearing subsequent step data. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.pip, self.steps, self.app_name)
        form = await request.form()
        step_id = form.get('step_id')
        pipeline_id = wand.db.get('pipeline_id', 'unknown')
@@ -500,7 +505,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        2. Revert Phase: Shows completed view with revert option
        3. Input Phase: Shows input form for new/updated value
        """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.pip, self.steps, self.app_name)
        step_id = 'step_01' # This string literal will be replaced
by swap_workflow_step.py
        step_index = self.steps_indices[step_id]
        step = steps[step_index] # Use the resolved step object
@@ -563,7 +568,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        await self.message_queue.add(pip, explanation,
verbatim=True)
        return Div(
            Card(
-                H3(f'{self.ui["EMOJIS"] ["USER_INPUT"] }
{wand.fmt(step.id)}: Enter {step.show}'),
+                H3(f'{self.ui["EMOJIS"] ["USER_INPUT"] }
{self.wand.fmt(step.id)}: Enter {step.show}'),
                P(explanation, cls='text-muted'),
                Label(
                    'Your Name:',
@@ -619,7 +624,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        3. Updates the workflow state
        4. Returns a UI showing the completed step and triggering the
next step
        """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.pip, self.steps, self.app_name)
        step_id = 'step_01' # This string literal will be replaced
by swap_workflow_step.py
        step_index = self.steps_indices[step_id]
        step = steps[step_index] # Use the resolved step object
@@ -651,7 +656,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        # --- START_STEP_BUNDLE: step_02 ---
        async def step_02(self, request):
            """ Handles GET request for Step 2: Displays input form or

```

```

completed value. """
-         pip, steps, app_name = (self.pip, self.steps, self.app_name)
+         wand, steps, app_name = (self.pip, self.steps, self.app_name)
        step_id = 'step_02' # This string literal will be replaced
by swap_workflow_step.py
        step_index = self.steps_indices[step_id]
        step = steps[step_index] # Use the resolved step object
@@ -712,7 +717,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        await self.message_queue.add(pip, explanation,
verbatimim=True)
        return Div(
            Card(
-                H3(f'{self.ui["EMOJIS"] ["GREETING"]}')
{wand.fmt(step.id)}: Enter {step.show}'),
+                H3(f'{self.ui["EMOJIS"] ["GREETING"]}')
{self.wand.fmt(step.id)}: Enter {step.show}'),
                P(explanation, cls='text-muted'),
                Label(
                    'Hello Message:',
@@ -760,7 +765,7 @@ You're here to make the workflow concepts
accessible and help users understand t

        async def step_02_submit(self, request):
            """ Handles POST submission for Step 2: Validates, saves
state, returns navigation. """
-         pip, steps, app_name = (self.pip, self.steps, self.app_name)
+         wand, steps, app_name = (self.pip, self.steps, self.app_name)
        step_id = 'step_02' # This string literal will be replaced
by swap_workflow_step.py
        step_index = self.steps_indices[step_id]
        step = steps[step_index] # Use the resolved step object
(nix)
[mike@nixos:~/repos/pipulate]$

```

But I still have an error here:

Debugging the HelloFlow wand Instantiation

```

14:34:58 | INFO      | imports.server_logging | [🌐 NETWORK] GET
/hello_workflow | ID: d55c106c
ERROR:      2026-03-08 14:34:58,723 | Exception in ASGI application
Traceback (most recent call last):
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/uvicorn/
protocols/http/httptools_impl.py", line 416, in run_asgi
    result = await app( # type: ignore[func-returns-value]

```

```

File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/uvicorn/
middleware/proxy_headers.py", line 60, in __call__
    return await self.app(scope, receive, send)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/applications.py", line 107, in __call__
    await self.middleware_stack(scope, receive, send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/errors.py", line 186, in __call__
    raise exc
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/errors.py", line 164, in __call__
    await self.app(scope, receive, _send)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/base.py", line 191, in __call__
    with recv_stream, send_stream, collapse_excgroups():
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File
"/nix/store/flbw79qdmvzbdrfd93avy5a7d29m2vb-python3-3.12.12/lib/pytho
n3.12/contextlib.py", line 158, in __exit__
    self.gen.throw(value)
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/_utils.py", line 87, in collapse_excgroups
    raise exc
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/base.py", line 193, in __call__
    response = await self.dispatch_func(request, call_next)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/home/mike/repos/pipulate/server.py", line 4305, in dispatch
    response = await call_next(request)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/base.py", line 168, in call_next
    raise app_exc from app_exc.__cause__ or app_exc.__context__
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
e/middleware/base.py", line 144, in coro
    await self.app(scope, receive_or_disconnect, send_no_error)
File

```

```
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/cors.py", line 87, in __call__
    await self.app(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/sessions.py", line 85, in __call__
    await self.app(scope, receive, send_wrapper)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/exceptions.py", line 63, in __call__
    await wrap_app_handling_exceptions(self.app, conn)(scope, receive,
send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 53, in wrapped_app
    raise exc
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 716, in __call__
    await self.middleware_stack(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 736, in app
    await route.handle(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 290, in handle
    await self.app(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 78, in app
    await wrap_app_handling_exceptions(app, request)(scope, receive,
send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 53, in wrapped_app
    raise exc
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 75, in app
```

```
response = await f(request)
^^^^^^^^^^^^^^^^^^^^^

File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/fasthtml
/core.py", line 627, in _f
    if not resp: resp = await _wrap_call(f, req, sig.parameters)
                          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/fasthtml
/core.py", line 477, in _wrap_call
    return await _handle(f, **wreq)
           ^^^^^^^^^^^^^^^^^^^^^^^

File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/fasthtml
/core.py", line 258, in _handle
    return (await f(*args, **kwargs)) if is_async_callable(f) else
await run_in_threadpool(f, *args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "/home/mike/repos/pipulate/server.py", line 4720, in home_route
    return await home(request)
           ^^^^^^^^^^^^^^^^^^

File "/home/mike/repos/pipulate/server.py", line 2049, in home
    response = await create_outer_container(current_profile_id, menux,
request, grayscale_enabled)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^

File "/home/mike/repos/pipulate/server.py", line 2543, in
create_outer_container
    Div(await create_grid_left(menux, request),
create_chat_interface(), cls='main-grid'),
      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "/home/mike/repos/pipulate/server.py", line 2621, in
create_grid_left
    content_to_render = await workflow_instance.landing(request)
                        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "/home/mike/repos/pipulate/apps/040_hello_workflow.py", line
335, in landing
    return wand.create_standard_landing_page(self)
           ^^^^^

NameError: name 'wand' is not defined. Did you mean: 'self.wand'?
```

unpacking signature (pip = self.pip instead of the tuple unpack).

Here is the surgical chisel-strike to fix the landing method in apps/040_hello_workflow.py:

```
    async def landing(self, request):
        """Generate the landing page using the standardized helper
        while maintaining WET explicitness."""
        # THE FIX: Change pip to wand
        wand = self.wand

        # Use centralized landing page helper - maintains WET
        principle by explicit call
        return wand.create_standard_landing_page(self)
```

Change that pip = self.pip to wand = self.wand and your Hello World app will be back online and fully integrated with the new wand mental model!

MikeLev.in: And that was just the start of it. Look at this:

```
(nix)
[mike@nixos:~/repos/pipulate]$ vim apps/040_hello_workflow.py
(nix)
[mike@nixos:~/repos/pipulate]$ git --no-pager diff
diff --git a/apps/040_hello_workflow.py b/apps/040_hello_workflow.py
index 5d23b2ce..31600af3 100644
--- a/apps/040_hello_workflow.py
+++ b/apps/040_hello_workflow.py
@@ -274,14 +274,20 @@ You're here to make the workflow concepts
accessible and help users understand t
    """
    self.app = app
    self.app_name = app_name
-    self.pip = pipulate
+    self.wand = pipulate
+    wand = self.wand
+
+    # THE FIX: Instantiate the wand!
+    self.wand = pipulate
+    wand = self.wand
+
    self.pipeline = pipeline
    self.steps_indices = {}
-    pip = self.pip
    self.message_queue = self.wand.get_message_queue()

    # Access centralized UI constants through dependency
    injection
-    self.ui = wand.get_ui_constants()
+    # THE FIX: Add 'self.' here, as 'wand' isn't a global import
+    self.ui = self.wand.get_ui_constants()

    # Define workflow steps
```

```

        # splice_workflow_step.py inserts new data steps BEFORE
        STEPS_LIST_INSERTION_POINT.
@@ -318,20 +324,20 @@ You're here to make the workflow concepts
accessible and help users understand t
        for step in self.steps:
            if step.id != 'finalize':
                self.step_messages[step.id] = {
-                    'input': f'{self.ui["EMOJIS"] ["INPUT_FORM"]}'
{wand.fmt(step.id)}: Please enter {step.show}.',
+                    'input': f'{self.ui["EMOJIS"] ["INPUT_FORM"]}'
{self.wand.fmt(step.id)}: Please enter {step.show}.',
                    'complete': f'{self.ui["EMOJIS"] ["SUCCESS"]}'
{step.show} complete. Continue to next step.'
                }

        async def landing(self, request):
            """Generate the landing page using the standardized helper
while maintaining WET explicitness."""
-            pip = self.pip
+            wand = self.wand

            # Use centralized landing page helper - maintains WET
principle by explicit call
            return wand.create_standard_landing_page(self)

        async def init(self, request):
            """ Handles the key submission, initializes state, and
renders the step UI placeholders. """
-            pip, steps, app_name = (self.pip, self.steps, self.app_name)
+            wand, steps, app_name = (self.wand, self.steps,
self.app_name)
            form = await request.form()
            user_input = form.get('pipeline_id', '').strip()
            if not user_input:
@@ -359,17 +365,17 @@ You're here to make the workflow concepts
accessible and help users understand t
                is_finalized = 'finalize' in state and 'finalized' in
state['finalize']

            # Progressive feedback with emoji conventions
-            await self.message_queue.add(pip,
f'{self.ui["EMOJIS"] ["WORKFLOW"]} Workflow ID: {pipeline_id}',
verbatim=True, spaces_before=0)
-            await self.message_queue.add(pip,
f'{self.ui["EMOJIS"] ["KEY"]} Return later by selecting '{pipeline_id}'
from the dropdown.', verbatim=True, spaces_before=0)
+            await self.message_queue.add(wand,
f'{self.ui["EMOJIS"] ["WORKFLOW"]} Workflow ID: {pipeline_id}',

```

```

verbatim=True, spaces_before=0)
+         await self.message_queue.add(wand,
f'{self.ui["EMOJIS"]["KEY"]} Return later by selecting '{pipeline_id}'
from the dropdown.', verbatim=True, spaces_before=0)

        if all_steps_complete:
            if is_finalized:
                status_msg = f'{self.ui["EMOJIS"]["LOCKED"]} Workflow
is complete and finalized. Use {self.ui["BUTTON_LABELS"]["UNLOCK"]} to
make changes.'
            else:
                status_msg = f'{self.ui["EMOJIS"]["SUCCESS"]}
Workflow is complete but not finalized. Press Finalize to lock your
data.'
-         await self.message_queue.add(pip, status_msg,
verbatim=True)
+         await self.message_queue.add(wand, status_msg,
verbatim=True)
        elif not any((step.id in state for step in self.steps)):
-         await self.message_queue.add(pip,
f'{self.ui["EMOJIS"]["INPUT_FORM"]} Please complete each step in
sequence. Your progress will be saved automatically.', verbatim=True)
+         await self.message_queue.add(wand,
f'{self.ui["EMOJIS"]["INPUT_FORM"]} Please complete each step in
sequence. Your progress will be saved automatically.', verbatim=True)

        parsed = wand.parse_pipeline_key(pipeline_id)
        prefix = f'{parsed['profile_part']}-{parsed['plugin_part']}-'
@@ -382,7 +388,7 @@ You're here to make the workflow concepts
accessible and help users understand t

    async def finalize(self, request):
        """ Handles GET request to show Finalize button and POST
request to lock the workflow. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.wand, self.steps,
self.app_name)
        pipeline_id = wand.db.get('pipeline_id', 'unknown')
        finalize_step = steps[-1]
        finalize_data = wand.get_step_data(pipeline_id,
finalize_step.id, {})
@@ -448,20 +454,20 @@ You're here to make the workflow concepts
accessible and help users understand t
    )

    else:
        await wand.finalize_workflow(pipeline_id)
-        await self.message_queue.add(pip,
self.step_messages['finalize']['complete'], verbatim=True)

```

```

+         await self.message_queue.add(wand,
self.step_messages['finalize']['complete'], verbatim=True)
        return wand.run_all_cells(app_name, steps)

    async def unfinalize(self, request):
        """ Handles POST request to unlock the workflow. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.wand, self.steps,
self.app_name)
        pipeline_id = wand.db.get('pipeline_id', 'unknown')
        await wand.unfinalize_workflow(pipeline_id)
-        await self.message_queue.add(pip,
f'{self.ui["EMOJIS"] ["UNLOCKED"]} Workflow unfinalized! You can now
revert to any step and make changes.', verbatim=True)
+        await self.message_queue.add(wand,
f'{self.ui["EMOJIS"] ["UNLOCKED"]} Workflow unfinalized! You can now
revert to any step and make changes.', verbatim=True)
        return wand.run_all_cells(app_name, steps)

    async def get_suggestion(self, step_id, state):
        """ Gets a suggested input value for a step, often using the
previous step's transformed output. """
-        pip, steps = (self.pip, self.steps)
+        wand, steps = (self.wand, self.steps)
        step = next((s for s in steps if s.id == step_id), None)
        if not step or not step.transform:
            return ''
@@ -475,19 +481,19 @@ You're here to make the workflow concepts
accessible and help users understand t

    async def handle_revert(self, request):
        """ Handles POST request to revert to a previous step,
clearing subsequent step data. """
-        pip, steps, app_name = (self.pip, self.steps, self.app_name)
+        wand, steps, app_name = (self.wand, self.steps,
self.app_name)
        form = await request.form()
        step_id = form.get('step_id')
        pipeline_id = wand.db.get('pipeline_id', 'unknown')
        if not step_id:
-            await self.message_queue.add(pip,
f'{self.ui["EMOJIS"] ["ERROR"]} Error: No step specified',
verbatim=True)
+            await self.message_queue.add(wand,
f'{self.ui["EMOJIS"] ["ERROR"]} Error: No step specified',
verbatim=True)
        return P('Error: No step specified', cls='text-invalid')
        await wand.clear_steps_from(pipeline_id, step_id, steps)

```

```

        state = wand.read_state(pipeline_id)
        state['_revert_target'] = step_id
        wand.write_state(pipeline_id, state)
        message = await wand.get_state_message(pipeline_id, steps,
self.step_messages)
-        await self.message_queue.add(pip,
f'{self.ui["EMOJIS"] ["WARNING"]} Reverted to {step_id}. {message}',
verbatim=True)
+        await self.message_queue.add(wand,
f'{self.ui["EMOJIS"] ["WARNING"]} Reverted to {step_id}. {message}',
verbatim=True)
        return wand.run_all_cells(app_name, steps)

# --- START_STEP_BUNDLE: step_01 ---
@@ -500,7 +506,7 @@ You're here to make the workflow concepts
accessible and help users understand t
    2. Revert Phase: Shows completed view with revert option
    3. Input Phase: Shows input form for new/updated value
    """
-    pip, steps, app_name = (self.pip, self.steps, self.app_name)
+    wand, steps, app_name = (self.wand, self.steps,
self.app_name)
        step_id = 'step_01' # This string literal will be replaced
by swap_workflow_step.py
        step_index = self.steps_indices[step_id]
        step = steps[step_index] # Use the resolved step object
@@ -514,7 +520,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        # Phase 1: Finalize Phase - Show locked view
        if 'finalized' in finalize_data:
            locked_msg = f'{self.ui["EMOJIS"] ["LOCKED"]} Your name is
set to: {user_val}'
-            await self.message_queue.add(pip, locked_msg,
verbatim=True)
+            await self.message_queue.add(wand, locked_msg,
verbatim=True)
        return Div(
            Card(
                H3(f'{self.ui["EMOJIS"] ["LOCKED"]} {step.show}:
{user_val}'),
@@ -535,7 +541,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        # Phase 2: Revert Phase - Show completed view with revert
option
        elif user_val and state.get('_revert_target') != step_id:
            completed_msg = f'{self.ui["EMOJIS"] ["SUCCESS"]} Step 1
is complete. You entered: {user_val}'
-            await self.message_queue.add(pip, completed_msg,

```

```

verbatim=True)
+         await self.message_queue.add(wand, completed_msg,
verbatim=True)
        return Div(
            wand.display_revert_header(
                step_id=step_id,
@@ -557,13 +563,13 @@ You're here to make the workflow concepts
accessible and help users understand t
        else:
            display_value = user_val if step.refill and user_val else
await self.get_suggestion(step_id, state)
            form_msg = f'{self.ui["EMOJIS"]["INPUT_FORM"]} Showing
name input form. No name has been entered yet.'
-         await self.message_queue.add(pip, form_msg,
verbatim=True)
-         await self.message_queue.add(pip,
self.step_messages[step_id]['input'], verbatim=True)
+         await self.message_queue.add(wand, form_msg,
verbatim=True)
+         await self.message_queue.add(wand,
self.step_messages[step_id]['input'], verbatim=True)
            explanation = f"Workflows are Notebooks without having to
look at the code. Let's collect some data..."
-         await self.message_queue.add(pip, explanation,
verbatim=True)
+         await self.message_queue.add(wand, explanation,
verbatim=True)
        return Div(
            Card(
-                 H3(f'{self.ui["EMOJIS"]["USER_INPUT"]}
{wand.fmt(step.id)}: Enter {step.show}'),
+                 H3(f'{self.ui["EMOJIS"]["USER_INPUT"]}
{self.wand.fmt(step.id)}: Enter {step.show}'),
                    P(explanation, cls='text-muted'),
                    Label(
                        'Your Name:',
@@ -619,7 +625,7 @@ You're here to make the workflow concepts
accessible and help users understand t
                3. Updates the workflow state
                4. Returns a UI showing the completed step and triggering the
next step
            """
-         pip, steps, app_name = (self.pip, self.steps, self.app_name)
+         wand, steps, app_name = (self.wand, self.steps,
self.app_name)
            step_id = 'step_01' # This string literal will be replaced
by swap_workflow_step.py
            step_index = self.steps_indices[step_id]

```

```

        step = steps[step_index] # Use the resolved step object
@@ -631,7 +637,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        # Validate input with emoji error handling
        if not user_val:
            error_msg = f'{self.ui["EMOJIS"]["ERROR"]} Please enter a
value'
-            await self.message_queue.add(self.pip, error_msg,
verbatim=True)
+            await self.message_queue.add(self.wand, error_msg,
verbatim=True)
            return P(error_msg, cls='text-invalid')

        # Update state
@@ -639,7 +645,7 @@ You're here to make the workflow concepts
accessible and help users understand t

        # Progressive feedback with emoji
        success_msg = f'{self.ui["EMOJIS"]["SUCCESS"]} Name saved:
{user_val}'
-        await self.message_queue.add(self.pip, success_msg,
verbatim=True)
+        await self.message_queue.add(self.wand, success_msg,
verbatim=True)

        # Update LLM context
        self.wand.append_to_history(f"[WIDGET CONTENT]
{step.show}:\n{user_val}")
@@ -651,7 +657,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        # --- START_STEP_BUNDLE: step_02 ---
        async def step_02(self, request):
            """ Handles GET request for Step 2: Displays input form or
completed value. """
-            pip, steps, app_name = (self.pip, self.steps, self.app_name)
+            wand, steps, app_name = (self.wand, self.steps,
self.app_name)
            step_id = 'step_02' # This string literal will be replaced
by swap_workflow_step.py
            step_index = self.steps_indices[step_id]
            step = steps[step_index] # Use the resolved step object
@@ -665,7 +671,7 @@ You're here to make the workflow concepts
accessible and help users understand t
            # Phase 1: Finalize Phase - Show locked view
            if 'finalized' in finalize_data:
                locked_msg = f'{self.ui["EMOJIS"]["LOCKED"]} Greeting is
locked: {user_val}'
-                await self.message_queue.add(pip, locked_msg,

```

```

verbatim=True)
+         await self.message_queue.add(wand, locked_msg,
verbatim=True)
        return Div(
            Card(
                H3(f'{self.ui["EMOJIS"] ["LOCKED"]} {step.show}:
{user_val}')),
@@ -686,7 +692,7 @@ You're here to make the workflow concepts
accessible and help users understand t
        # Phase 2: Revert Phase - Show completed view with revert
option
        elif user_val and state.get('_revert_target') != step_id:
            completed_msg = f'{self.ui["EMOJIS"] ["SUCCESS"]} Step 2
is complete. Greeting: {user_val}'
-         await self.message_queue.add(pip, completed_msg,
verbatim=True)
+         await self.message_queue.add(wand, completed_msg,
verbatim=True)
        return Div(
            wand.display_revert_header(
                step_id=step_id,
@@ -707,12 +713,12 @@ You're here to make the workflow concepts
accessible and help users understand t
        # Phase 3: Input Phase - Show input form
        else:
            display_value = user_val if step.refill and user_val else
await self.get_suggestion(step_id, state)
-         await self.message_queue.add(pip,
self.step_messages[step_id] ['input'], verbatim=True)
+         await self.message_queue.add(wand,
self.step_messages[step_id] ['input'], verbatim=True)
            explanation = f"That's it! Workflows just collect data -
walking you from one Step to the Next Step ▶"
-         await self.message_queue.add(pip, explanation,
verbatim=True)
+         await self.message_queue.add(wand, explanation,
verbatim=True)
        return Div(
            Card(
-                 H3(f'{self.ui["EMOJIS"] ["GREETING"]}
{wand.fmt(step.id)}: Enter {step.show}'),
+                 H3(f'{self.ui["EMOJIS"] ["GREETING"]}
{self.wand.fmt(step.id)}: Enter {step.show}'),
                P(explanation, cls='text-muted'),
                Label(
                    'Hello Message:',
@@ -760,7 +766,7 @@ You're here to make the workflow concepts
accessible and help users understand t

```

```

    async def step_02_submit(self, request):
        """ Handles POST submission for Step 2: Validates, saves
        state, returns navigation. """
        -         pip, steps, app_name = (self.pip, self.steps, self.app_name)
        +         wand, steps, app_name = (self.wand, self.steps,
        self.app_name)
            step_id = 'step_02' # This string literal will be replaced
by swap_workflow_step.py
            step_index = self.steps_indices[step_id]
            step = steps[step_index] # Use the resolved step object
@@ -776,7 +782,7 @@ You're here to make the workflow concepts
accessible and help users understand t
            is_valid, error_msg, error_component =
wand.validate_step_input(user_val, step.show)
            if not is_valid:
                error_with_emoji = f'{self.ui["EMOJIS"]["ERROR"]}'
{error_msg}'
        -         await self.message_queue.add(pip, error_with_emoji,
verbatim=True)
        +         await self.message_queue.add(wand, error_with_emoji,
verbatim=True)
            return error_component

        processed_val = user_val
@@ -784,10 +790,10 @@ You're here to make the workflow concepts
accessible and help users understand t

        # Progressive feedback with emoji
        success_msg = f'{self.ui["EMOJIS"]["SUCCESS"]} {step.show}:
{processed_val}'
        -         await self.message_queue.add(pip, success_msg, verbatim=True)
        +         await self.message_queue.add(wand, success_msg,
verbatim=True)

        if wand.check_finalize_needed(step_index, steps):
        -         await self.message_queue.add(pip,
self.step_messages['finalize']['ready'], verbatim=True)
        +         await self.message_queue.add(wand,
self.step_messages['finalize']['ready'], verbatim=True)

        return self.wand.chain_reverter(
            step_id=step_id,
diff --git a/foo_files.py b/foo_files.py
index c1f11820..d99b4a8a 100644
--- a/foo_files.py
+++ b/foo_files.py
@@ -15,7 +15,7 @@

```

```
# 2. foo_files.py # <-- this

AI_PHOOEY_CHOP = """\
-server.py
+# server.py
  apps/040_hello_workflow.py

# foo_files.py
(nix)
[mike@nixos:~/repos/pipulate]$
```

Ugh, these workflows aren't going to be so straight forward to get into the wand style. Some of my WET workflows are really long. I might be able to come up with a very controlled Python search and replace program that could do this. At least I got Hello World working again. I've still got some errors in python server.py I have to look into. But at least I took care of this one.

Addressing Lingering Startup Errors

Gemini 3 Pro: Context verified.

Beautiful diff. You successfully banked the win on the Hello World workflow, and seeing that clean Git history update feels great.

You bring up the fundamental trade-off of the WET (Write Explicitly, Terminate) philosophy. The immense benefit is that each workflow is completely decoupled, transparent, and easy for an amnesiac AI to read top-to-bottom. The cost, however, is exactly what you are feeling right now: when you make a foundational semantic shift—like upgrading the pip mental model to the wand mental model—the surface area for that change is huge.

Your instinct to build a controlled Python search-and-replace script is spot on. Doing this manually across dozens of long workflows is a recipe for fatigue and human error.

We have a couple of great ways to tackle that automation when you're ready:

1. **The Command-Line Crucible:** We could craft a highly surgical `rg | xargs sed` pipeline (leveraging your existing command-line flow state) to handle the highly predictable replacements.
2. **The AST Approach:** If the WET files have enough slight variations that regex feels too dangerous, we can write a quick Python script using the `ast` module to programmatically find the `pip`, `steps`, `app_name` = `(self.pip, self.steps, self.app_name)` unpacking patterns and safely rewrite them to `wand`.

Whenever you're ready, let me know where you want to point the flashlight. Do we want to forge that automated refactoring script to sweep through the rest of the apps/ directory, or do you want to drop the logs for those lingering server.py errors and slay some more bugs?

MikeLev.in: Help me track down this error:

```
(nix)
[mike@nixos:~/repos/pipulate]$ python server.py
✔ Data directory ensured at top of script.
```

```

|
|
|-----|
| / _ \ / _ / / / / _ \ |
| / / / / / / / / / / _ / |
| / / , _ / / / / _ / ____ /
| / _ / | _ / _ \ ____ / _ /
|
|
starting...

```

Pipulate server

| - Curious? Become a Carpenter!
 \) ""\ | |

$$\left| \begin{array}{cc} _ > - _ > \\ _ & \end{array} \right|$$

Down the rabbit
hole of radical transparency

Pipulate Architecture

```
graph LR; subgraph Navigation; direction TB; N1[Navigation] --- N2[Search, Profiles, Apps, Settings]; end; Chat[Chat] --- Workflow[Workflow]; Workflow --> MainArea[Main Area]; AppUI[App UI] --- LLM[LLM Interaction]; LLM --> Interface[Interface]; MainArea --- Pipeline["(Pipeline)"];
```

← Search, Profiles,

Apps, Settings

Workflow, Main Area

App UI | (Pipeline)

Interface $\longleftarrow$ LLM Interaction

```
15:02:20 | INFO      | imports.server_logging | 🚀 FINDER_TOKEN:
UNIFIED_LOGGING_ACTIVE - Single source of truth logging initialized
15:02:20 | INFO      | imports.server_logging | 📁 FINDER_TOKEN:
```


|_ |

Pipulate

Workflows and CRUD Apps

```
15:02:20 | INFO      | imports.durable_backup_system | 🗄 Rolling
backup root: /home/mike/.pipulate/backups
✅ Data directory ensured at top of script.
15:02:20 | INFO      | imports.server_logging | 🚀 FINDER_TOKEN:
UNIFIED_LOGGING_ACTIVE - Single source of truth logging initialized
15:02:20 | INFO      | imports.server_logging | 📁 FINDER_TOKEN:
LOG_ROTATION_READY - Keeping last 10 server runs for debugging context
15:02:20 | INFO      | server                | 🚀 FINDER_TOKEN: EARLY_STARTUP
- Logger initialized, beginning server startup sequence
15:02:20 | INFO      | server                | 🏷 FINDER_TOKEN: APP_CONFIG -
App name: Botifython
15:02:20 | INFO      | server                | 🗄 FINDER_TOKEN: DB_CONFIG -
Database filename: data/botifython_dev.db
15:02:20 | INFO      | server                | 🤖 FINDER_TOKEN: LLM_CONFIG -
Model: gemma3, Max words: 80, Conversation length: 600, Context
window: 128k tokens
15:02:20 | INFO      | server                | SSE Broadcaster initialized
15:02:20 | INFO      | server                | 🗄 FINDER_TOKEN: DB_WRAPPER -
Database wrapper initialized
15:02:20 | INFO      | pipulate.core         | Pipulate initializing in
server mode.
15:02:20 | INFO      | server                | 🗄 FINDER_TOKEN: PIPULATE -
Pipeline object created.
15:02:20 | INFO      | server                | 🌐 FINDER_TOKEN:
CORS_MIDDLEWARE - CORS middleware added to FastHTML app
15:02:20 | INFO      | server                | 📁 FINDER_TOKEN: PLUGINS_DIR -
Plugins directory exists
15:02:20 | INFO      | server                | 💬 FINDER_TOKEN: CHAT_INIT -
Chat instance initialized
15:02:20 | INFO      | server                | 🔗 FINDER_TOKEN: CHAT_LINK -
Chat reference set in pipulate instance
15:02:20 | ERROR     | server                | Error importing plugin module
440_browser_automation: No module named 'pkg_resources'
15:02:20 | WARNING   | server                | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class roles.PluginIdentityManager
```

```

appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping
15:02:20 | WARNING | server | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class tasks.PluginIdentityManager
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping
15:02:20 | INFO | apps.030_roles | Fastlite 'roles' table created
or accessed via handle: "roles"
15:02:20 | INFO | apps.030_roles | Called .dataclass() on table
handle to enable dataclass returns.
15:02:20 | INFO | imports.crud | 🗄 roles CRUD initialized with
durable backup support
15:02:20 | INFO | apps.030_roles | ROLES: Role initialization
complete.
15:02:20 | INFO | apps.060_tasks | Fastlite 'tasks' table created
or accessed via handle: "tasks"
15:02:20 | INFO | apps.060_tasks | Called .dataclass() on table
handle to enable dataclass returns.
15:02:20 | INFO | imports.crud | 🗄 tasks CRUD initialized with
durable backup support
15:02:20 | INFO | apps.060_tasks | 🔍 FINDER_TOKEN:
STARTUP_TASKS_BASELINE - No pending tasks found across all profiles
15:02:20 | INFO | server | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION - Dynamically discovering all MCP tools
Error setting up paths or importing dependencies: 'tools' directory
not found at expected location: /home/mike/repos/tools
15:02:21 | INFO | server | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION_COMPLETE - 19 tools are now available
15:02:21 | INFO | server | 🔧 ASCII_MARKER_1: About to
check MCP figlet banner conditions
15:02:21 | INFO | server | 🔧 ASCII_MARKER_1: __name__ =
server, tool_count = 19
15:02:21 | INFO | server | 🔧 ASCII_MARKER_X: MCP figlet
banner conditions not met - skipped
15:02:21 | INFO | server | 🔧 STARTUP_MARKER_1: About to
check __name__ == '__main__' condition
15:02:21 | INFO | server | 🔧 STARTUP_MARKER_1: __name__
value is: server
15:02:21 | INFO | server | 🔧 STARTUP_MARKER_X: NOT in
__name__ == '__main__' block - Rich tables skipped
😊 Chip O'Theseus: All plugins loaded successfully! The workshop is
fully equipped.
15:02:21 | INFO | server | 🔧 TRANSPARENCY_MARKER_1:
About to check radical transparency banner conditions
15:02:21 | INFO | server | 🔧 TRANSPARENCY_MARKER_1:
__name__ = server
15:02:21 | INFO | server | 🔧 TRANSPARENCY_MARKER_X:
radical transparency banner conditions not met - skipped

```

```

15:02:21 | ERROR      | __main__          | Error importing plugin module
440_browser_automation: No module named 'pkg_resources'
15:02:21 | WARNING   | __main__          | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class
profiles.ProfilesPluginIdentity appears to be a plugin (has
APP_NAME/DISPLAY_NAME/ROLES/steps) but missing required landing method
- skipping
15:02:21 | WARNING   | __main__          | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class roles.PluginIdentityManager
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping
15:02:21 | WARNING   | __main__          | FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class tasks.PluginIdentityManager
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping
15:02:21 | INFO      | __main__          | Instantiating ProfilesPlugin
with profiles_table_from_server
15:02:21 | INFO      | apps.020_profiles | 👤 Profiles Plugin SUCCESS:
Initialized with 'profiles' table object: profile
15:02:21 | INFO      | imports.crud      | 📦 profiles CRUD initialized
with durable backup support
15:02:21 | INFO      | apps.020_profiles | CRUD routes for 👤 Profiles
(prefix '/profiles') registered by ProfileCrudOperations.
15:02:21 | INFO      | apps.030_roles    | Fastlite 'roles' table created
or accessed via handle: "roles"
15:02:21 | INFO      | apps.030_roles    | Called .dataclass() on table
handle to enable dataclass returns.
15:02:21 | INFO      | imports.crud      | 📦 roles CRUD initialized with
durable backup support
15:02:21 | INFO      | apps.030_roles    | ROLES: Role initialization
complete.
15:02:21 | INFO      | apps.060_tasks    | Fastlite 'tasks' table created
or accessed via handle: "tasks"
15:02:21 | INFO      | apps.060_tasks    | Called .dataclass() on table
handle to enable dataclass returns.
15:02:21 | INFO      | imports.crud      | 📦 tasks CRUD initialized with
durable backup support
15:02:21 | INFO      | apps.060_tasks    | 🔍 FINDER_TOKEN:
STARTUP_TASKS_BASELINE - No pending tasks found across all profiles
15:02:21 | INFO      | __main__          | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION - Dynamically discovering all MCP tools
15:02:21 | INFO      | __main__          | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION_COMPLETE - 19 tools are now available
15:02:21 | INFO      | __main__          | 🔧 ASCII_MARKER_1: About to
check MCP figlet banner conditions
15:02:21 | INFO      | __main__          | 🔧 ASCII_MARKER_1: __name__ =
__main__, tool_count = 19
15:02:21 | INFO      | __main__          | 🔧 ASCII_MARKER_2: Displaying

```

[illegible]

$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ 0 & 1 \end{pmatrix}$

$\left| \begin{array}{c} | \\ |_{-}) \\ | \end{array} \right\rangle$




Model

MCP Arsenal

AI-powered tools for transparency	Equipped with 19
-----------------------------------	------------------

```
15:02:21 | INFO      | __main__      | 🔧 ASCII_MARKER_3: MCP figlet
banner and section header displayed
15:02:21 | INFO      | __main__      | 🔧 STARTUP_MARKER_1: About to
check __name__ == '__main__' condition
15:02:21 | INFO      | __main__      | 🔧 STARTUP_MARKER_1: __name__
value is: __main__
15:02:21 | INFO      | __main__      | 🔧 STARTUP_MARKER_2: Inside
__name__ == '__main__' block - showing Rich tables
```

Startup Complete  Pipulate

 System Status

Component	Status	Details
 App Name	 Active	Botifython
 Environment	 Active	Development
 Plugins	 Loaded	38/38 registered
 MCP Tools	 Ready	19 tools available
 AI Memory enabled	 Active	Keychain persistence
 Browser Eyes capability	 Ready	Session hijacking

🎯 AI Quick Commands

Purpose	Command
🔍 System State	<code>.venv/bin/python cli.py call pipeline_state_inspector</code>
📖 Log Analysis local_llm_grep_logs	<code>.venv/bin/python cli.py call --search_term FINDER_TOKEN</code>
👁 Browser Scrape browser_scrape_page	<code>.venv/bin/python cli.py call --url http://localhost:5001</code>
🤖 Session Hijack from	<code>.venv/bin/python -c "import asyncio; tools.mcp_tools import execute_complete_session_hijacking; asyncio.run(execute_complete_session_hijacking({}... </code>
🧠 AI Discovery tools.mcp_tools import	<code>.venv/bin/python -c "from ai_self_discovery_assistant; import asyncio; asyncio.run(ai_self_discovery_assistant({'discove... </code>
	<code>'capabilities'}}))"</code>

operational • Ready for AI workflows

```
15:02:21 | INFO      | __main__ | 🔧 STARTUP_MARKER_3:
aa.startup_summary_table displayed
```

Discovery Complete

AI Capabilities Discovered

Capability	Status	Description
 MCP Tools tools ready	 Active	19 programmatic
 Browser Eyes analysis	 Ready	Screenshot + DOM
 Browser Hands control	 Ready	Selenium automation
 Session Hijack capture	 Ready	Complete workflow
 Persistent Memory restarts	 Active	Keychain across
 Documentation guides	 Complete	14 comprehensive

Zero uncertainty •

100% automation success rate

15:02:21 | INFO | __main__ |  STARTUP_MARKER_4:

ai_breadcrumb_summary displayed

 AI

Assistant Guidelines

 Critical Environment Notes

Issue	Solution
-------	----------

 Python Path Issue 'venv/bin/python' (even	NEVER use 'python' - ALWAYS use
	with .venv in prompt!)
 Golden Path cli.py call <tool_name>	cd pipulate && .venv/bin/python
	works everywhere
 Working Directory - never run from	MCP tools require 'cd pipulate' first
	workspace root
 Browser Evidence FIRST, system internals	Check browser_cache/looking_at/
	second

Essential

knowledge for 100% success rate

```
15:02:21 | INFO      | __main__          | 🔧  STARTUP_MARKER_5:
aa.startup_environment_warnings displayed
💬 Chip O'Theseus: All plugins loaded successfully! The workshop is
fully equipped.
15:02:21 | INFO      | __main__          | 🔧  TRANSPARENCY_MARKER_1:
About to check radical transparency banner conditions
15:02:21 | INFO      | __main__          | 🔧  TRANSPARENCY_MARKER_1:
__name__ = __main__
15:02:21 | INFO      | __main__          | 🔧  TRANSPARENCY_MARKER_2:
Displaying radical transparency banner
```

RADICAL TRANSPARENCY

Every operation is observable • Every call is logged

Every state change is tracked • Every error is explained

🔍 FINDER_TOKENs guide your debugging journey

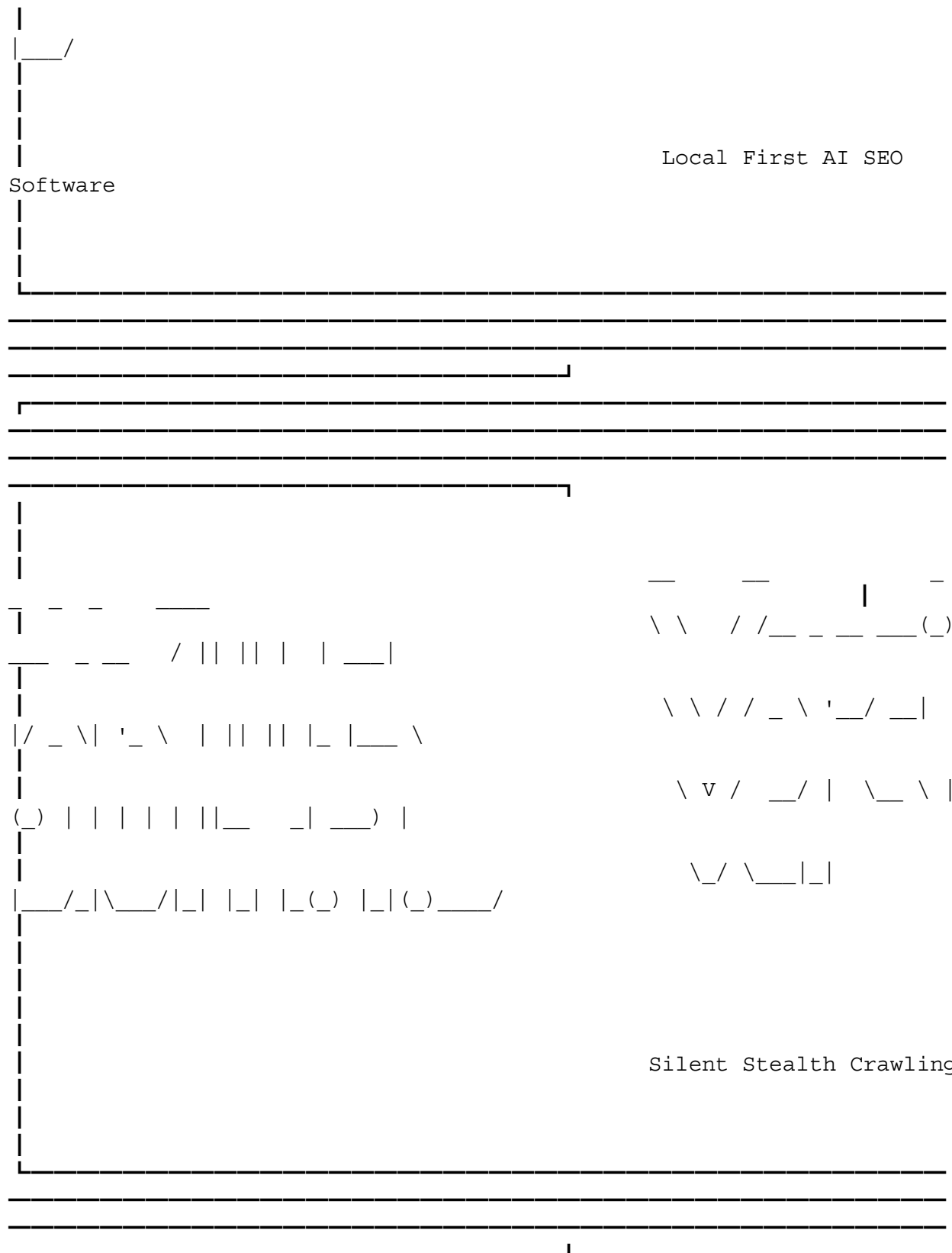
🔧 MCP Tools provide programmatic access to everything

📊 Pipeline State Inspector reveals workflow internals

🤖 AI Assistants have complete system visibility

"Know EVERYTHING that's happening!"

```
15:02:21 | INFO      | __main__          | 🔧  TRANSPARENCY_MARKER_3:
aa.radical_transparency_banner displayed
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE:
WATCHDOG RESTART BEHAVIOR EXPLANATION
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE:
The Pipulate server uses a sophisticated dual-display system:
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE: 1.
CONSOLE (for humans): Shows each banner ONCE per session for clean UX
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE: 2.
LOGS (for AI assistants): Captures ALL restart events including rapid
restarts
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE:
When watchdog detects file changes, multiple rapid restarts may occur:
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE: -
Each restart triggers aa.figlet_banner() calls
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE: -
Console shows banners once (better newb/plebe experience)
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE: -
Logs capture every restart event for AI transparency
15:02:21 | WARNING   | __main__          | 🤖  AI_RESTART_ARCHITECTURE: -
This creates a cleaner user experience while maintaining radical
transparency for AI
```

💬 Chip O'Theseus: Hello! The server is restarting. I'll be right back

online.

15:02:21 | INFO | __main__ | 🌐 FINDER_TOKEN: ENVIRONMENT -
Current environment: Development

15:02:21 | WARNING | imports.server_logging | [⚠️ WARNING]

Development mode active | Using database: data/botifython_dev.db

15:02:22 | INFO | imports.server_logging | [🚀 STARTUP] Server
starting on http://localhost:5001

15:02:22 | INFO | __main__ | 🌐 FINDER_TOKEN: UVICORN_START
- Starting uvicorn server on http://localhost:5001

15:02:22 | INFO | __main__ | 📊 FINDER_TOKEN:
UVICORN_CONFIG - Log level: warning, Access log: False

15:02:22 | INFO | __main__ | 🔧 FINDER_TOKEN:

STARTUP_EVENT_MCP_READY - 19 MCP tools available for async startup

15:02:22 | INFO | __main__ | SERVER STARTUP_EVENT: Pre
synchronize_roles_to_db.

🔒 Synchronizing roles and permissions

15:02:22 | INFO | __main__ | SYNC_ROLES: Starting role
synchronization to database...

15:02:22 | INFO | __main__ | SYNC_ROLES: Total unique role
names discovered from all sources: {'Core', 'Tutorial', 'Developer',
'Components', 'Workshop', 'Botify Employee'}

15:02:22 | INFO | __main__ | SYNC_ROLES: Synchronization
complete. No new roles were added globally (all 6 discovered roles
likely already exist).

15:02:22 | INFO | __main__ | SERVER STARTUP_EVENT: Post
synchronize_roles_to_db. Final startup states:

15:02:22 | INFO | __main__ | 💬 FINDER_TOKEN:
CONVERSATION_RESTORE_STARTUP - Conversation history restoration
disabled to prevent database locking conflicts

15:02:22 | INFO | __main__ | 💬 FINDER_TOKEN:
CONVERSATION_RESTORE_NONE - Starting with fresh conversation history
(SQLite locking prevention)

15:02:22 | INFO | __main__ | 🧩 FINDER_TOKEN:
DEMO_CONTINUATION_NONE - No demo continuation state found

15:02:22 | INFO | __main__ | 🧩 FINDER_TOKEN:

DEMO_RESTART_NONE - Normal server restart, no demo state found

15:02:22 | INFO | imports.durable_backup_system | 📁 Rolling
backup root: /home/mike/.pipulate/backups

📖 Backup System

Protecting critical data assets

15:02:22 | INFO | imports.durable_backup_system | 🛡️ Latest backup
created: /home/mike/.pipulate/backups/ai_keychain.db

15:02:22 | INFO | imports.durable_backup_system | 🛡️ Dated backup
already exists: /home/mike/.pipulate/backups/ai_keychain_2026-03-08.db

15:02:22 | INFO | imports.durable_backup_system | 🛡️ Latest backup
created: /home/mike/.pipulate/backups/discussion.db

15:02:22 | INFO | imports.durable_backup_system | 🛡️ Dated backup
already exists: /home/mike/.pipulate/backups/discussion_2026-03-08.db

```
15:02:22 | INFO      | imports.durable_backup_system | 🛡️ Latest backup
created: /home/mike/.pipulate/backups/botifython.db
15:02:22 | INFO      | imports.durable_backup_system | 🛡️ Dated backup
already exists: /home/mike/.pipulate/backups/botifython_2026-03-08.db
15:02:22 | INFO      | imports.durable_backup_system | 🛡️ Latest backup
created: /home/mike/.pipulate/backups/botifython_dev.db
15:02:22 | INFO      | imports.durable_backup_system | 🛡️ Dated backup
already exists:
/home/mike/.pipulate/backups/botifython_dev_2026-03-08.db
15:02:22 | INFO      | imports.durable_backup_system | 🧹 Cleanup
complete: No old backup files found (>7 days)
15:02:22 | INFO      | imports.durable_backup_system | 🛡️ Database
backup complete: 4/4 successful
15:02:22 | INFO      | __main__          | 🛡️ FINDER_TOKEN:
BACKUP_STARTUP_SUCCESS - All 4 databases backed up successfully
📖 Workshop Ready
```

All systems initialized and ready for creative exploration

🛡️ Durable Backup Status

Database	Status	Backup Path
Production Profiles/Tasks	✅ Backed Up (44.0 KB)	/home/mike/.pipulate/backups/botifython.db
Development Profiles/Tasks	✅ Backed Up (312.0 KB)	/home/mike/.pipulate/backups/botifython_dev.db
Conversation History	✅ Backed Up (7232.0 KB)	/home/mike/.pipulate/backups/discussion.db
Chip O'Theseus Memory	✅ Backed Up (28.0 KB)	/home/mike/.pipulate/backups/ai_keychain.db

```
15:02:22 | INFO      | imports.server_logging | 📊 STARTUP - RICH TABLE: 🛡️ Durable Backup Status
Headers: Database | Status | Backup Path
```

🔑 AI Keychain Memory

Key	Value
alias_test	it works
idea	foo bar baz

```
15:02:22 | INFO      | imports.server_logging |
```

 STARTUP - RICH TABLE:  AI Keychain Memory
Headers: Key | Value

 Database State: db

```
{
  "last_app_choice": "hello_workflow",
  "last_visited_url": "/hello_workflow",
  "last_profile_id": "1",
  "pipeline_id": "Default_Profile-hello-28",
  "current_environment": "Development",
  "profile_locked": "0",
  "split-sizes": "[65, 35]",
  "theme_preference": "dark",
  "intro_current_page": "1",
  "voice_enabled": "0",
  "endpoint_message_sent_hello_workflow_Development": "sent"
}
```

15:02:22 | INFO | imports.server_logging |  FINDER_TOKEN:
DB_STATE_DB - STARTUP FINAL Key-Value Store: [Rich JSON displayed to console]

15:02:22 | INFO | imports.server_logging |  SEMANTIC_DB_DB:
STARTUP FINAL Human-readable state:

 Current app/workflow: hello_workflow

 Last page visited: /hello_workflow

 Active user profile: 1

 Environment mode: Development

 Profile editing:  Unlocked

 UI layout split: [65, 35]

 UI theme: dark

 Startup message sent for

endpoint_message_sent_hello_workflow_Development: sent

 Table State: profiles

```
[
  {
    "id": 1,
    "name": "Default Profile",
    "real_name": "Default User",
    "address": "",
    "code": "",
    "active": 1,
    "priority": 0
  }
]
```

15:02:22 | INFO | imports.server_logging |  FINDER_TOKEN:

TABLE_STATE_PROFILES - STARTUP FINAL Snapshot: [Rich JSON displayed to console]

```

 Total workflows: 47
 Finalized: 1
 Active: 46
 Apps: browser(2), hello(28), notebook(12), param_buster(2),
selenium_url_opener(2), widgets(1)

```

```
🕒 Recent activity (24h): 3 workflows
15:02:22 | INFO      | imports.server_logging | 🔍
SEMANTIC_PIPELINE_CONTEXT: STARTUP FINAL Recent workflow keys:
Default_Profile-hello-26, Default_Profile-hello-27,
Default_Profile-hello-28
```

```
15:02:24 | INFO      | __main__ |  SSE event loop reference
stored for restart notifications
```

```
15:02:24 | INFO      | imports.server_logging | [ NETWORK] GET
/hello workflow | ID: ed587e79
```

```
15:02:24 | INFO      | imports.server_logging | [ NETWORK] GET  
/hello workflow | ID: 95a10eb3
```

```
ERROR:      2026-03-08 15:02:25,044 | Exception in ASGI application
Traceback (most recent call last):
```

```
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/uvicorn/
protocols/websockets/websockets_impl.py", line 244, in run_asgi
    result = await self.app(self.scope, self.asgi_receive,
self.asgi_send) # type: ignore[func-returns-value]
```

[illegible]

```
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/applications.py", line 107, in __call__
    await self.middleware_stack(scope, receive, send)
```

```
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/errors.py", line 151, in __call__
    await self.app(scope, receive, send)
```

```
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/base.py", line 103, in __call__
    await self.app(scope, receive, send)
```

```
File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlett
```

```
e/middleware/cors.py", line 79, in __call__
    await self.app(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/sessions.py", line 85, in __call__
    await self.app(scope, receive, send_wrapper)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/exceptions.py", line 63, in __call__
    await wrap_app_handling_exceptions(self.app, conn)(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 53, in wrapped_app
    raise exc
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 716, in __call__
    await self.middleware_stack(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 736, in app
    await route.handle(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 364, in handle
    await self.app(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 97, in app
    await wrap_app_handling_exceptions(app, session)(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 53, in wrapped_app
    raise exc
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 95, in app
    await func(session)
```

File

```
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/fasthtml  
/live_reload.py", line 30, in live_reload_ws  
    while True: await websocket.receive()  
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

File

```
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/websockets.py", line 57, in receive
```

```
    raise RuntimeError('Cannot call "receive" once a disconnect  
message has been received.')  
RuntimeError: Cannot call "receive" once a disconnect message has been  
received.
```

```
15:02:25 | INFO      | imports.server_logging | [🌐 NETWORK] GET  
/demo-bookmark-check | ID: 78bd7605  
15:02:25 | INFO      | __main__              | 📖 No demo bookmark found  
15:02:25 | INFO      | imports.server_logging | [🌐 NETWORK] GET  
/check-demo-resume | ID: 41951200  
15:02:25 | INFO      | imports.server_logging | [🌐 NETWORK] GET  
/check-demo-comeback | ID: aa9c0857  
15:02:25 | INFO      | __main__              | 🐞 DEBUG: /check-demo-comeback  
called, demo_comeback_state in db: None  
15:02:25 | INFO      | imports.server_logging | [🌐 NETWORK] GET  
/hello_workflow | ID: b648092a  
15:02:25 | INFO      | pipulate.core         | [🔄 QUEUEING] 🚀 Server  
started in DEV mode. Ready for experimentation and testing!...  
15:02:25 | INFO      | __main__              | 💬 FINDER_TOKEN:  
MESSAGE_APPENDED - ID:1, Role:system, Content:🚀 Server started in DEV  
mode. Ready for experiment...  
15:02:25 | WARNING   | pipulate.core         | Deleted key from persistence  
store: endpoint_message_sent_hello_workflow_Development  
15:02:25 | INFO      | __main__              | 🔧 STARTUP_DEBUG: Initial  
last_app_choice='hello_workflow', last_visited_url='/hello_workflow'  
15:02:25 | INFO      | __main__              | 🔧 STARTUP_DEBUG: URL endpoint  
resolved: /hello_workflow -> 'hello_workflow'  
15:02:25 | INFO      | __main__              | 🔧 STARTUP_DEBUG: Final  
current_endpoint='hello_workflow' (empty string = homepage)  
15:02:25 | INFO      | __main__              | 🔧 STARTUP_DEBUG: Available  
plugin_instances: ['introduction', 'backup_introduction', 'profiles',  
'aspect', 'roles', 'hello_workflow', 'documentation', 'tasks',  
'history', 'connect_with_botify', 'parameter_buster', 'link_graph',  
'gap_analysis', 'workflow_genesis', 'widget_examples', 'roadmap',  
'dev_assistant', 'simon_mcp', 'blank_placeholder', 'botify_trifecta',  
'stream_simulator', 'text_field', 'text_area', 'dropdown',  
'checkboxes', 'radios', 'range', 'switch', 'upload', 'markdown',  
'mermaid', 'prism', 'javascript', 'pandas', 'rich', 'matplotlib',  
'webbrowser', 'selenium']  
15:02:25 | INFO      | __main__              | 💬 FINDER_TOKEN:  
MESSAGE_APPENDED - ID:2, Role:system, Content:# Workflow Template
```

Your Role

Reading Guide

```
Log Format: TIME | LEVEL | MODULE | MESSAGE
```

Common Emojis & Meanings:

- | | | |
|---|----------|--|
|  | STARTUP | - Server initialization and startup events |
|  | NETWORK | - HTTP requests, API calls, web traffic |
|  | PIPELINE | - Workflow execution and step processing |
|  | DATABASE | - Data storage operations and queries |
|  | PROFILE | - User profile and authentication events |
|  | PLUGIN | - Plugin loading and workflow registration |

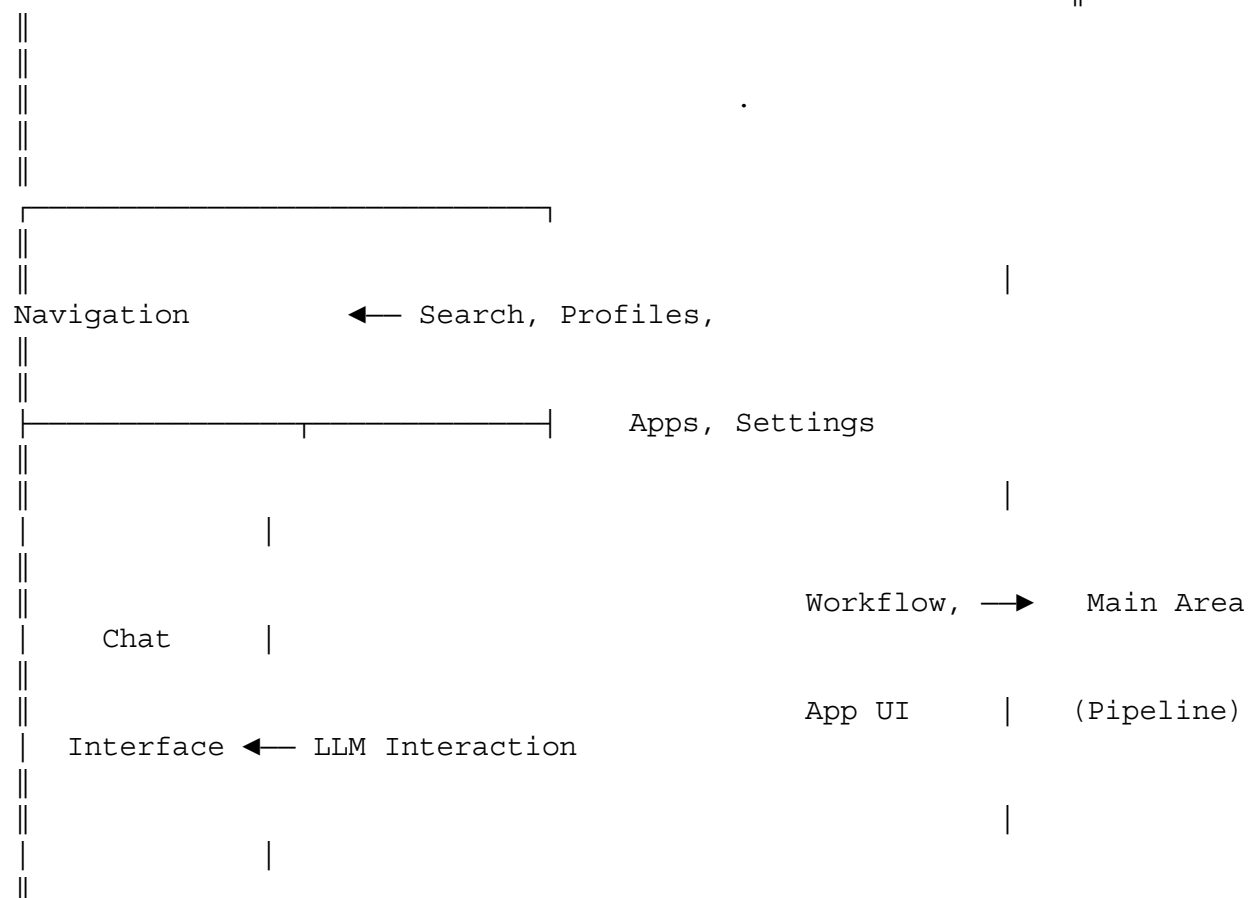
	💬	CHAT	- LLM interactions and AI conversations
	🤖	AI_CREATIVE	- ASCII art and AI-specific logging (logs only)
You're speaking			
	🔍	FINDER_TOKEN	- Searchable debug markers for AI assistants
my language!			
	🔧	MCP_TOOLS	- Model Context Protocol tool operations
,	0		
	🌐	BROWSER	- Browser automation and Selenium operations
\\	.	0	
	🎯	SUCCESS	- Completion markers and achievements
\\	o		
	🏷️	CONFIG	- System configuration and tagging
/	"	'\	
	🗄️	DB_CONFIG	- Database configuration events
.	.	.	
	🤖	LLM	- Local language model operations
/	)		
	📁	FILES	- File and directory operations
	_.'		
	🧹	CLEANUP	- Housekeeping and maintenance
-'	/	\	
	✨	FRESH	- New state creation and refresh
	🍞	BREADCRUMBS	- AI discovery guidance (AI_BREADCRUMB_01-04)
	📸	CAPTURE	- Screenshots and visual state
	📝	INPUT	- Form inputs and user data entry
	📁	UPLOAD	- File upload operations
	✅	COMPLETE	- Task completion and success
	⚠️	WARNING	- Important notices and potential issues
	❌	ERROR	- System errors and failures

Pro Tips:

- Use grep with emojis: `grep "🌐" logs/server.log`
- Find specific events: `grep "FINDER_TOKEN" logs/server.log`
- Follow live logs: `tail -f logs/server.log`
- AI breadcrumbs: `grep "AI_BREADCRUMB" logs/server.log`
- MCP tool calls: `grep "MCP_.*_START" logs/server.log`

Understanding what you're seeing in the logs

Pipulate Architecture



about dependencies), it's failing to import. **The Fix:** You have two choices:

- **Modernize (Recommended):** Open `apps/440_browser_automation.py` and replace `import pkg_resources` with Python's built-in `import importlib.metadata`.
- **Or Patch:** Add `setuptools` to your `requirements.in` / `flake.nix`.

2. The Rogue tools Directory Path

The Log: Error setting up paths or importing dependencies: 'tools' directory not found at expected location: `/home/mike/repos/tools` **The Issue:** One of your tool files (likely `tools/__init__.py` or `tools/mcp_tools.py`) has a `Path(__file__)` resolution that is going one directory too high. It's looking for tools directly inside `repos/` instead of `repos/pipulate/`. **The Fix:** Check the imports/setup in your `tools/` directory. You likely have a `.parent.parent` that just needs to be a `.parent`, or a `sys.path.insert` that is resolving one level too high.

3. The PluginIdentityManager Log Spam

The Log: `FINDER_TOKEN: PLUGIN_REGISTRATION_FAILURE - Plugin class roles.PluginIdentityManager appears to be a plugin... but missing required landing method (also happens for tasks and profiles)` **The Issue:** Your `find_plugin_classes()` heuristic in `server.py` is beautifully explicit, but it's a little too greedy. It sees your helper classes (`PluginIdentityManager`) have attributes like `APP_NAME` and `ROLES`, assumes they are malformed plugins, and throws a warning. **The Fix:** Let's add a quick filter in `server.py` to ignore your known structural base/helper classes.

Here is the chisel-strike for `server.py` (around line 552):

```
        else:
            # Only log classes that look like they might
            # be plugins (have common plugin attributes)
            if any(hasattr(obj, attr) for attr in
                ['APP_NAME', 'DISPLAY_NAME', 'ROLES', 'steps']):
                # THE FIX: Ignore helper/base classes to
                # stop log spam
                if not any(ignore in name for ignore in
                    ['Identity', 'Manager', 'Base']):
                    logger.warning(f'FINDER_TOKEN:
                    PLUGIN_REGISTRATION_FAILURE - Plugin class {module_name}.{name}
                    appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
                    missing required landing method - skipping')
```

Knock those three out, and your `server.py` startup sequence will be absolutely pristine! Want to tackle the tools path resolution together, or are you ready to look at that automated Python search-and-replace script for the WET workflows?

MikeLevin: Okay, but I don't see `pkg_resources`:

```
(nix)
[mike@nixos:~/repos/pipulate]$ rg pkg_resources
pipulate/pipulate.py
13:# Suppress the specific UserWarning from pkg_resources in
selenium-wire
17:     message="pkg_resources is deprecated as an API.*"
```

```

server.py
101:warnings.filterwarnings("ignore", category=UserWarning,
message=".*pkg_resources.*")
248:warnings.filterwarnings("ignore", category=UserWarning,
message=".*pkg_resources.*")
(nix)
[mike@nixos:~/repos/pipulate]$

```

And I did this:

```

(nix)
[mike@nixos:~/repos/pipulate]$ git --no-pager diff
diff --git a/foo_files.py b/foo_files.py
index 828ea83d..363b162f 100644
--- a/foo_files.py
+++ b/foo_files.py
@@ -15,7 +15,8 @@
 # 2. foo_files.py # <-- this

AI_PHOOEY_CHOP = """\
-server.py
+# server.py
+tools/__init__.py
# apps/040_hello_workflow.py

# foo_files.py
diff --git a/server.py b/server.py
index dd8a83bb..c41454f2 100644
--- a/server.py
+++ b/server.py
@@ -1657,7 +1657,9 @@ def find_plugin_classes(plugin_modules,
discovered_modules):
    else:
        # Only log classes that look like they might
be plugins (have common plugin attributes)
        if any(hasattr(obj, attr) for attr in
['APP_NAME', 'DISPLAY_NAME', 'ROLES', 'steps']):
-            logger.warning(f'FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class {module_name}.{name}
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but
missing required landing method - skipping')
+            # THE FIX: Ignore helper/base classes to
stop log spam
+            if not any(ignore in name for ignore in
['Identity', 'Manager', 'Base']):
+                logger.warning(f'FINDER_TOKEN:
PLUGIN_REGISTRATION_FAILURE - Plugin class {module_name}.{name}
appears to be a plugin (has APP_NAME/DISPLAY_NAME/ROLES/steps) but

```

```

missing required landing method - skipping')
    except Exception as e:
        logger.error(f'FINDER_TOKEN: PLUGIN_REGISTRATION_FAILURE
- Error processing module {module_or_name}: {str(e)}')
        import traceback

(nix)
[mike@nixos:~/repos/pipulate]$

```

MikeLev.in: Interesting! I gave up selenium-wire which I was only using to get HTTP request headers like you can with the Requests library. Eventually the W3C BiDi protocol will let us do that formally, but it's not widely supported yet. But what I really ended up doing was switching to undetected-chromedriver which is mutually exclusive with selenium-wire but the benefits were so vast I dumped selenium-wire so I can purge it out and shouldn't be getting that error anymore. I may have left cruft.

Gemini 3 Pro: Context verified.

Ah, the phantom dependency! You nailed it. selenium-wire relies heavily on blinker and pyOpenSSL, which historically dragged pkg_resources into the party. You didn't import it; selenium-wire snuck it in through the back door.

Ripping out selenium-wire is a massive win anyway. It's heavy, intercepts all traffic through a local proxy (which breaks frequently), and as you mentioned, undetected-chromedriver combined with the upcoming W3C BiDi standard is the far superior modern path.

You've got two pieces of cruft to slay.

1. Purge selenium-wire from Browser Automation

In apps/440_browser_automation.py, you have lingering selenium-wire imports and request-inspection logic in Step 2.

Remove the import at the top (around line 17):

```

# DELETE THIS LINE:
from seleniumwire import webdriver as wire_webdriver

```

Fix step_02_submit (around line 316): Replace the wire_webdriver call and rip out the driver.requests loop (since standard Selenium doesn't have it). We'll stub the status and headers for now until you wire up BiDi.

```

    # CHANGE THIS:
    # driver = wire_webdriver.Chrome(service=service,
options=chrome_options)
    driver = webdriver.Chrome(service=service,
options=chrome_options)

    await self.message_queue.add(pip, f'Crawling URL with
Selenium: {url}', verbatim=True)
    driver.get(url)
    await asyncio.sleep(2)
    title = driver.title
    source = driver.page_source
    dom = driver.execute_script('return
document.documentElement.outerHTML;')

```

```

        # RIP OUT THE ENTIRE `driver.requests` LOOP AND REPLACE
WITH STUBS:
        headers = {"Notice": "selenium-wire removed; headers
unavailable"}
        status = 200

        domain, path = get_safe_path(url)

```

2. Fix the Rogue Path in LLM Optics

In tools/llm_optics.py, you have a classic "I moved the file but didn't update the relative path math" bug. The comments even give it away: # Assumes script is in Notebooks/imports. You moved it to tools/!

Fix the path resolution (around line 29):

```

# --- Path Configuration (Robust sys.path setup) ---
try:
    # THE FIX: Adjust the `.parent` traversal for the new location
    script_dir = Path(__file__).resolve().parent # This is the
'tools' dir
    project_root = script_dir.parent             # This is the
'pipulate' root
    tools_dir = script_dir

    # if not tools_dir.is_dir(): ... (leave the rest as-is)

```

Make those two surgical strikes, and that beautiful startup log is going to be completely error-free. pkg_resources will be banished to the shadow realm, and your tools directory will map perfectly. Ready to strike?

MikeLev.in: Here's the first edit.

```

(nix)
[mike@nixos:~/repos/pipulate]$ vim apps/440_browser_automation.py
(nix)
[mike@nixos:~/repos/pipulate]$ git --no-pager diff
diff --git a/apps/440_browser_automation.py
b/apps/440_browser_automation.py
index b07ef9c7..118c2a61 100644
--- a/apps/440_browser_automation.py
+++ b/apps/440_browser_automation.py
@@ -14,7 +14,6 @@ from selenium.webdriver.chrome.service import
Service
    from selenium.webdriver.common.by import By
    from selenium.webdriver.support import expected_conditions as EC
    from selenium.webdriver.support.ui import WebDriverWait
-from seleniumwire import webdriver as wire_webdriver
    from starlette.responses import HTMLResponse, JSONResponse
    from webdriver_manager.chrome import ChromeDriverManager

```

```

@@ -350,7 +349,7 @@ class BrowserAutomation:
    return Div(Card(H3('Crawl URL'), Form(Input(type='url',
name='url', placeholder='https://example.com', required=True,
value=display_value, cls='contrast'), Button('Crawl URL',
type='submit', cls='primary'),
hx_post=f'/{app_name}/{step_id}_submit', hx_target=f'#{step_id}')),
Div(id=next_step_id), id=step_id)

    async def step_02_submit(self, request):
-        """Process the Crawl URL submission, open with Selenium-wire,
and save crawl data."""
+        """Process the Crawl URL submission, open with Selenium, and
save crawl data."""
        pip, steps, app_name = (self.pipulate, self.steps,
self.app_name)
        step_id = 'step_02'
        step_index = self.steps_indices[step_id]
@@ -381,7 +380,7 @@ class BrowserAutomation:
        await self.message_queue.add(pip, 'Using system
Chrome for Linux', verbatim=True)
        service = Service()
        await self.message_queue.add(pip, 'Initializing Chrome
driver...', verbatim=True)
-        driver = wire_webdriver.Chrome(service=service,
options=chrome_options)
+        driver = webdriver.Chrome(service=service,
options=chrome_options)
        await self.message_queue.add(pip, f'Crawling URL with
Selenium: {url}', verbatim=True)
        driver.get(url)
        await asyncio.sleep(2)
@@ -389,21 +388,11 @@ class BrowserAutomation:
        source = driver.page_source
        dom = driver.execute_script('return
document.documentElement.outerHTML;')
        main_request = None
-        for request in driver.requests:
-            if request.response and request.url.startswith(url):
-                main_request = request
-                break
-        if not main_request:
-            for request in driver.requests:
-                if request.response:
-                    main_request = request
-                    break
-        if main_request and main_request.response:
-            headers = dict(main_request.response.headers)

```

```

-             status = main_request.response.status_code
-         else:
-             headers = {}
-             status = 200
+
+         # RIP OUT THE ENTIRE `driver.requests` LOOP AND REPLACE
WITH STUBS:
+         headers = {"Notice": "selenium-wire removed; headers
unavailable"}
+         status = 200
+
+         domain, path = get_safe_path(url)
+         date_slug = datetime.now().strftime('%Y%m%d')
+         base_dir = ensure_crawl_dir(app_name, domain, date_slug)
(nix)
[mike@nixos:~/repos/pipulate]$

```

And the other half:

```

(nix)
[mike@nixos:~/repos/pipulate]$ git commit -am "Stripping
selenium-wire"
[main a774f021] Stripping selenium-wire
1 file changed, 7 insertions(+), 18 deletions(-)
(nix)
[mike@nixos:~/repos/pipulate]$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 48 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 531 bytes | 531.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
remote: This repository moved. Please use the new location:
remote:  git@github.com:pipulate/pipulate.git
To github.com:miklevin/pipulate
bb23f4c5..a774f021  main -> main
(nix)
[mike@nixos:~/repos/pipulate]$ vim tools/llm_optics.py
(nix)
[mike@nixos:~/repos/pipulate]$ git --no-pager diff
diff --git a/tools/llm_optics.py b/tools/llm_optics.py
index 9b388785..cbbfce55 100644
--- a/tools/llm_optics.py
+++ b/tools/llm_optics.py
@@ -43,9 +43,10 @@
@@ -43,9 +43,10 @@
CONSOLE_WIDTH = 180

# --- Path Configuration (Robust sys.path setup) ---
try:

```



```
e/middleware/cors.py", line 79, in __call__
    await self.app(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/sessions.py", line 85, in __call__
    await self.app(scope, receive, send_wrapper)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/middleware/exceptions.py", line 63, in __call__
    await wrap_app_handling_exceptions(self.app, conn)(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 53, in wrapped_app
    raise exc
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 716, in __call__
    await self.middleware_stack(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 736, in app
    await route.handle(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 364, in handle
    await self.app(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 97, in app
    await wrap_app_handling_exceptions(app, session)(scope, receive, send)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 53, in wrapped_app
    raise exc
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/_exception_handler.py", line 42, in wrapped_app
    await app(scope, receive, sender)
  File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/routing.py", line 95, in app
    await func(session)
```

```

File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/fasthtml
/live_reload.py", line 30, in live_reload_ws
    while True: await websocket.receive()
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File
"/home/mike/repos/pipulate/.venv/lib/python3.12/site-packages/starlette/websockets.py", line 57, in receive
    raise RuntimeError('Cannot call "receive" once a disconnect
message has been received.')
RuntimeError: Cannot call "receive" once a disconnect message has been
received.
15:43:55 | INFO          | imports.server_logging | [🌐 NETWORK] GET
/demo-bookmark-check | ID: aa7708de

```

Gemini 3 Pro: Context verified.

Brilliant surgical strikes! You successfully excised selenium-wire (and took pkg_resources down with it) and fixed the directory traversal in llm_optics.py. Your repo is shedding technical debt rapidly.

As for that final RuntimeError in the logs—I have good news. **That is a completely harmless phantom.** Look closely at the traceback path: .../fasthtml/core.py -> .../starlette/websockets.py -> .../fasthtml/live_reload.py

Here is exactly what's happening: Because you have live=True set in your fast_app() instantiation in server.py, FastHTML automatically injects a hidden WebSocket connection into your browser to listen for Hot Module Reload (HMR) pings.

When your Watchdog triggers a server restart (or when you quickly manually refresh the browser), the browser abruptly severs that live-reload WebSocket connection. In the microsecond before the old server process completely dies, the fasthtml/live_reload.py background task tries to run await websocket.receive() one last time. It hits the severed connection, Starlette throws a RuntimeError, and Uvicorn dramatically logs it as an ERROR before quietly dying.

Notice how right below the error, the network requests continue seamlessly ([🌐 NETWORK] GET /demo-bookmark-check)? That's the *new* server instance already up and running perfectly.

How to handle it:

1. **The Zen Path (Recommended):** Just ignore it. It's a known, benign artifact of ASGI frameworks (Starlette/FastAPI) running in dev mode with live-reloading. It only happens when a client disconnects abruptly during a reload.
2. **The Mute Path:** If the visual noise bothers you, you can change live=True to live=False in server.py (around line 588), but you'll lose FastHTML's built-in auto-refresh triggers.

Since your server is now booting clean (minus the harmless HMR phantom), we have a completely stable runway.

Are you ready to forge that automated Python script to mass-migrate your WET workflows from pip to wand? If so, I can draft up an AST-based (or Regex-based) refactoring script that you can run from your scripts/ directory to safely upgrade all the apps/* files in one fell swoop.

MikeLev.in: Okay. That is a rabbit hole I can ignore for a bit. Isn't there just some pause I could insert somewhere?

Gemini 3 Pro: Context verified.

I love the instinct—when a system is tripping over its own feet during a transition, adding a tiny

asyncio.sleep() is usually the perfect shock absorber.

But in this specific case, a pause won't save you.

Here is why: Uvicorn's Hot Module Reload watchdog acts like an assassin. When you save a file, it doesn't politely ask the server to finish its current loop. It aggressively terminates the worker process and spins up a new one. That RuntimeError is essentially the Starlette websocket letting out a death rattle as the floor is violently pulled out from underneath it. If you put a pause in there, the process just gets killed in its sleep instead of while listening at the door—but the broken pipe error still registers on the way down.

However, since you're already doing some elegant "vegetable eating" with your logging architecture, you can apply a silencer to Uvicorn so it stops screaming about this specific, harmless event.

You can slip this custom filter right into server.py near where you configure your other loggers (around the warnings.filterwarnings lines):

```
import logging

class MuteHMRPhantom(logging.Filter):
    """Silences the harmless FastHTML live-reload disconnect error."""
    def filter(self, record):
        if record.exc_info:
            exc_type, exc_value, _ = record.exc_info
            if exc_type is RuntimeError and 'Cannot call "receive"
once a disconnect' in str(exc_value):
                return False
        return True

# Apply the gag to Uvicorn's error logger
logging.getLogger("uvicorn.error").addFilter(MuteHMRPhantom())
```

That acts as a perfectly tailored earplug. The server will still restart instantly, FastHTML will still live-reload your browser, but the console will remain beautifully pristine.

Shall we snap that silencer onto server.py, or would you like to pivot to forging that automated Python script to mass-replace pip with wand across your entire apps/ directory?

MikeLev.in:

```
Traceback (most recent call last):
  File "/home/mike/repos/pipulate/server.py", line 103, in <module>
    class MuteHMRPhantom(logging.Filter):
                        ^^^^^^^
NameError: name 'logging' is not defined. Did you forget to import
'logging'?
(nix)
[mike@nixos:~/repos/pipulate]$
```

Gemini 3 Pro: Context verified.

Ah, the classic missing import! We handed you the silencer but forgot to give you the thread adapter to screw it on.

You just need to add import logging right above that new class in server.py.

Here is the exact chisel-strike:

```
warnings.filterwarnings("ignore", category=UserWarning,
```

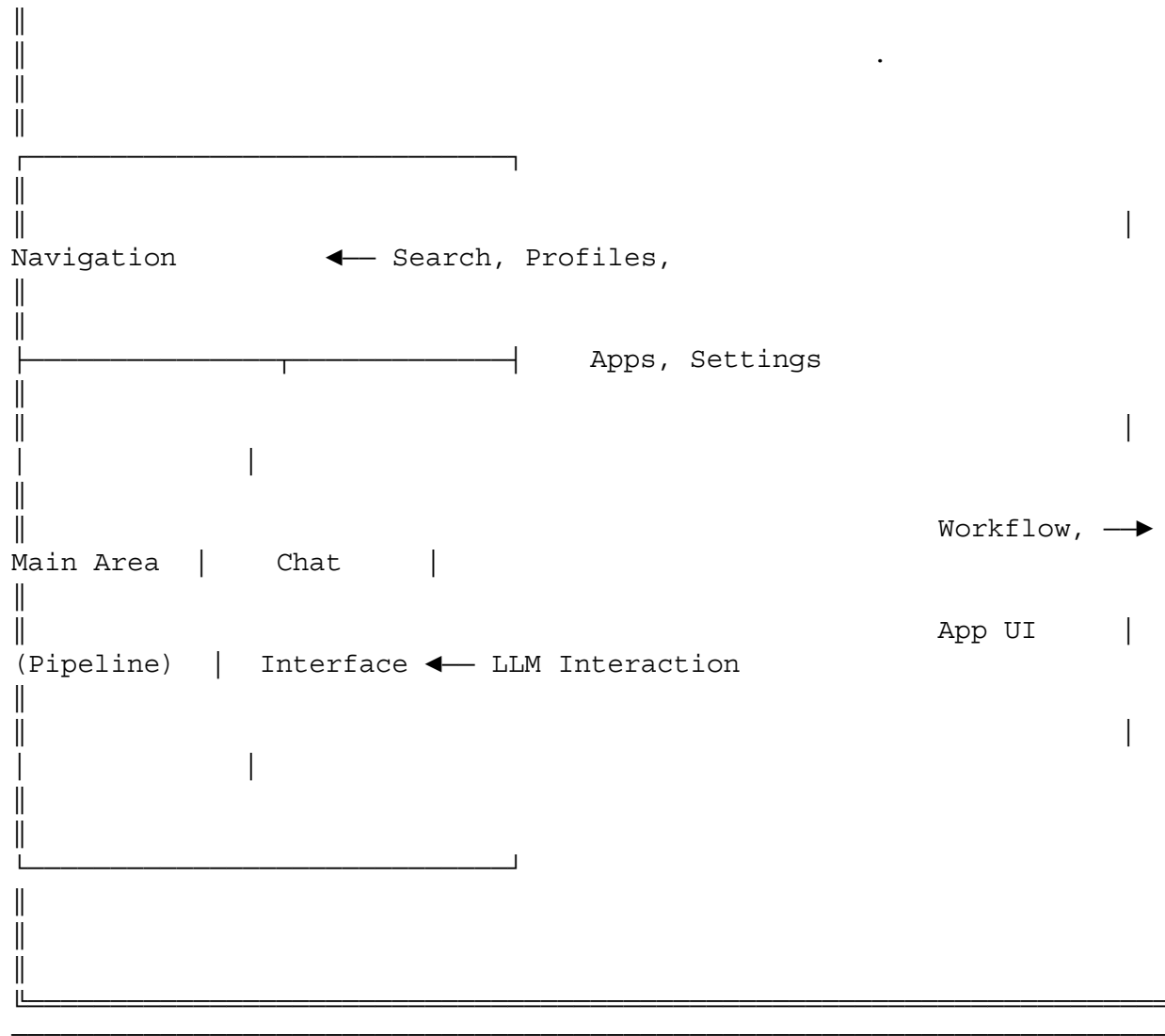

server starting... Pipulate

🐰 Welcome to Consoleland

webserver output. /) _____
(i.e. the Uvicorn
is talking to you) /) _ // / \
gobbledygook now ___ (/ _ 0 0 |
but over time it
will not. *(==(_T_)== |Pipulate|
Carpenter! \) "" \ |
>->_> _____/

Down the
rabbit hole of radical transparency

= 🛠 Pipulate Architecture



```

16:09:55 | INFO      | imports.server_logging | 🚀 FINDER_TOKEN:
UNIFIED_LOGGING_ACTIVE - Single source of truth logging initialized
16:09:55 | INFO      | imports.server_logging | 📁 FINDER_TOKEN:
LOG_ROTATION_READY - Keeping last 10 server runs for debugging context
16:09:55 | INFO      | __main__              | 🚀 FINDER_TOKEN: EARLY_STARTUP
- Logger initialized, beginning server startup sequence
16:09:55 | INFO      | __main__              | 🚀 Running in INFO mode (edit
server.py and set DEBUG_MODE=True for verbose logging)
16:09:55 | INFO      | __main__              | 🏷️ FINDER_TOKEN: APP_CONFIG -
App name: Botifython
16:09:55 | INFO      | __main__              | 📁 FINDER_TOKEN: DB_CONFIG -
Database filename: data/botifython_dev.db
16:09:55 | INFO      | __main__              | 🤖 FINDER_TOKEN: LLM_CONFIG -
Model: gemma3, Max words: 80, Conversation length: 600, Context
window: 128k tokens
16:09:55 | INFO      | __main__              | SSE Broadcaster initialized

```

```
16:09:55 | INFO      | imports.durable_backup_system | 🗄️ Rolling
backup root: /home/mike/.pipulate/backups
✅ Data directory ensured at top of script.
16:09:55 | INFO      | imports.server_logging | 🚀 FINDER_TOKEN:
UNIFIED_LOGGING_ACTIVE - Single source of truth logging initialized
16:09:55 | INFO      | imports.server_logging | 📁 FINDER_TOKEN:
LOG_ROTATION_READY - Keeping last 10 server runs for debugging context
16:09:55 | INFO      | server      | 🚀 FINDER_TOKEN: EARLY_STARTUP
- Logger initialized, beginning server startup sequence
16:09:55 | INFO      | server      | 🏷️ FINDER_TOKEN: APP_CONFIG -
App name: Botifython
16:09:55 | INFO      | server      | 🗄️ FINDER_TOKEN: DB_CONFIG -
Database filename: data/botifython_dev.db
16:09:55 | INFO      | server      | 🤖 FINDER_TOKEN: LLM_CONFIG -
Model: gemma3, Max words: 80, Conversation length: 600, Context
window: 128k tokens
16:09:55 | INFO      | server      | SSE Broadcaster initialized
16:09:55 | INFO      | server      | 🗄️ FINDER_TOKEN: DB_WRAPPER -
Database wrapper initialized
16:09:55 | INFO      | pipulate.core | Pipulate initializing in
server mode.
16:09:55 | INFO      | server      | 🗄️ FINDER_TOKEN: PIPULATE -
Pipeline object created.
16:09:55 | INFO      | server      | 🌐 FINDER_TOKEN:
CORS_MIDDLEWARE - CORS middleware added to FastHTML app
16:09:55 | INFO      | server      | 📁 FINDER_TOKEN: PLUGINS_DIR -
Plugins directory exists
16:09:55 | INFO      | server      | 💬 FINDER_TOKEN: CHAT_INIT -
Chat instance initialized
16:09:55 | INFO      | server      | 🔗 FINDER_TOKEN: CHAT_LINK -
Chat reference set in pipulate instance
16:09:56 | INFO      | apps.030_roles | Fastlite 'roles' table created
or accessed via handle: "roles"
16:09:56 | INFO      | apps.030_roles | Called .dataclass() on table
handle to enable dataclass returns.
16:09:56 | INFO      | imports.crud | 🗄️ roles CRUD initialized with
durable backup support
16:09:56 | INFO      | apps.030_roles | ROLES: Role initialization
complete.
16:09:56 | INFO      | apps.060_tasks | Fastlite 'tasks' table created
or accessed via handle: "tasks"
16:09:56 | INFO      | apps.060_tasks | Called .dataclass() on table
handle to enable dataclass returns.
16:09:56 | INFO      | imports.crud | 🗄️ tasks CRUD initialized with
durable backup support
16:09:56 | INFO      | apps.060_tasks | 🔍 FINDER_TOKEN:
```

```

STARTUP_TASKS_BASELINE - No pending tasks found across all profiles
16:09:56 | INFO      | server      | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION - Dynamically discovering all MCP tools
16:09:56 | INFO      | server      | 🔧 FINDER_TOKEN:
STARTUP_MCP_REGISTRATION_COMPLETE - 19 tools are now available
16:09:56 | INFO      | server      | 🔧 ASCII_MARKER_1: About to
check MCP figlet banner conditions
16:09:56 | INFO      | server      | 🔧 ASCII_MARKER_1: __name__ =
server, tool_count = 19
16:09:56 | INFO      | server      | 🔧 ASCII_MARKER_X: MCP figlet
banner conditions not met - skipped
16:09:56 | INFO      | server      | 🔧 STARTUP_MARKER_1: About to
check __name__ == '__main__' condition
16:09:56 | INFO      | server      | 🔧 STARTUP_MARKER_1: __name__
value is: server
16:09:56 | INFO      | server      | 🔧 STARTUP_MARKER_X: NOT in
__name__ == '__main__' block - Rich tables skipped
💬 Chip O'Theseus: All plugins loaded successfully! The workshop is
fully equipped.
16:09:56 | INFO      | server      | 🔧 TRANSPARENCY_MARKER_1:
About to check radical transparency banner conditions
16:09:56 | INFO      | server      | 🔧 TRANSPARENCY_MARKER_1:
__name__ = server
16:09:56 | INFO      | server      | 🔧 TRANSPARENCY_MARKER_X:
radical transparency banner conditions not met - skipped
16:09:56 | INFO      | __main__    | Instantiating ProfilesPlugin
with profiles_table_from_server
16:09:56 | INFO      | apps.020_profiles | 👤 Profiles Plugin SUCCESS:
Initialized with 'profiles' table object: profile
16:09:56 | INFO      | imports.crud | 📦 profiles CRUD initialized
with durable backup support
16:09:56 | INFO      | apps.020_profiles | CRUD routes for 👤 Profiles
(prefix '/profiles') registered by ProfileCrudOperations.
16:09:56 | INFO      | apps.030_roles | Fastlite 'roles' table created
or accessed via handle: "roles"
16:09:56 | INFO      | apps.030_roles | Called .dataclass() on table
handle to enable dataclass returns.
16:09:56 | INFO      | imports.crud | 📦 roles CRUD initialized with
durable backup support
16:09:56 | INFO      | apps.030_roles | ROLES: Role initialization
complete.
16:09:56 | INFO      | apps.060_tasks | Fastlite 'tasks' table created
or accessed via handle: "tasks"
16:09:56 | INFO      | apps.060_tasks | Called .dataclass() on table
handle to enable dataclass returns.
16:09:56 | INFO      | imports.crud | 📦 tasks CRUD initialized with
durable backup support
16:09:56 | INFO      | apps.060_tasks | 🔍 FINDER_TOKEN:

```

STARTUP_TASKS_BASELINE - No pending tasks found across all profiles

```
16:09:56 | INFO | __main__ |  FINDER_TOKEN:
```

STARTUP_MCP_REGISTRATION - Dynamically discovering all MCP tools

```
16:09:56 | INFO | main |  FINDER_TOKEN:
```

STARTUP_MCP_REGISTRATION_COMPLETE - 19 tools are now available

```
16:09:56 | INFO      | __main__ | 🛠️ ASCII_MARKER_1: About to
check MCP figlet banner conditions
```

```
16:09:56 | INFO      | __main__ |  ASCII_MARKER_1: __name__ =
__main__, tool_count = 19
```

```
16:09:56 | INFO      | __main__ | 🛠 ASCII_MARKER_2: Displaying
MCP figlet banner and section header
```

$\mid$ $\backslash$ $\mid$ $_$ $\mid$ $_$ $\backslash$

| | M | | | | _) |

| | | | | _ | _ /

|_ | |_ |\ _|_|

Model Context Protocol Tools

Equipped with 19

AI-powered tools for transparency

```
16:09:56 | INFO      | __main__      | 🔧 ASCII_MARKER_3: MCP figlet
banner and section header displayed
16:09:57 | INFO      | __main__      | 🔧 STARTUP_MARKER_1: About to
check __name__ == '__main__' condition
16:09:57 | INFO      | __main__      | 🔧 STARTUP_MARKER_1: __name__
value is: __main__
16:09:57 | INFO      | __main__      | 🔧 STARTUP_MARKER_2: Inside
__name__ == '__main__' block - showing Rich tables
```

 Pipulate Startup Complete System Status

Component	Status	Details
 App Name	 Active	Botifython
 Environment	 Active	Development

	 Plugins	 Loaded	39/39 registered
	 MCP Tools	 Ready	19 tools available
	 AI Memory enabled	 Active	Keychain persistence
	 Browser Eyes capability	 Ready	Session hijacking

	 AI Quick Commands	
--	---	--

	Purpose	Command	
--	---------	---------	--

	 System State	.venv/bin/python cli.py call	
		pipeline_state_inspector	
	 Log Analysis local_llm_grep_logs	.venv/bin/python cli.py call	
		--search_term FINDER_TOKEN	
	 Browser Scrape browser_scrape_page	.venv/bin/python cli.py call	
		--url http://localhost:5001	

```
| | 🧑🔧 Session Hijack | .venv/bin/python -c "import asyncio;
from |
| | |
| | | tools.mcp_tools import
| | |
| | | execute_complete_session_hijacking;
| | |
| | |
asyncio.run(execute_complete_session_hijacking({}... |
|
| | 🧠 AI Discovery | .venv/bin/python -c "from
tools.mcp_tools import |
| | |
| | | ai_self_discovery_assistant; import
asyncio; |
| | |
| | |
asyncio.run(ai_self_discovery_assistant({'discove... |
| | |
| | | 'capabilities'}}))"
```

operational • Ready for AI workflows

```
16:09:57 | INFO      | __main__ | 🛠️  STARTUP_MARKER_3:
aa.startup_summary_table displayed
```

🍞 AI Discovery Complete

AI Capabilities Discovered

Capability	Status	Description
 MCP Tools tools ready	 Active	19 programmatic
 Browser Eyes analysis	 Ready	Screenshot + DOM
 Browser Hands control	 Ready	Selenium automation
 Session Hijack capture	 Ready	Complete workflow
 Persistent Memory restarts	 Active	Keychain across
 Documentation guides	 Complete	14 comprehensive

Zero
uncertainty • 100% automation success rate

16:09:57 | INFO | __main__ |  STARTUP_MARKER_4:
ai_breadcrumb_summary displayed

 AI Assistant Guidelines

 Critical Environment Notes

Issue	Solution
-------	----------

 Python Path Issue '.venv/bin/python' (even	NEVER use 'python' - ALWAYS use with .venv in prompt!)
 Golden Path cli.py call <tool_name>	cd pipulate && .venv/bin/python works everywhere
 Working Directory - never run from	MCP tools require 'cd pipulate' first workspace root
 Browser Evidence FIRST, system internals	Check browser_cache/looking_at/ second

Essential knowledge for 100% success rate

```

16:09:57 | INFO      | __main__          |  STARTUP_MARKER_5:
aa.startup_environment_warnings displayed
😊 Chip O'Theseus: All plugins loaded successfully! The workshop is
fully equipped.
16:09:57 | INFO      | __main__          |  TRANSPARENCY_MARKER_1:
About to check radical transparency banner conditions
16:09:57 | INFO      | __main__          |  TRANSPARENCY_MARKER_1:
__name__ = __main__
16:09:57 | INFO      | __main__          |  TRANSPARENCY_MARKER_2:
Displaying radical transparency banner

```

RADICAL TRANSPARENCY

Every operation is observable • Every call is logged
Every state change is tracked • Every error is explained



FINDER_TOKENS guide your debugging journey



MCP Tools provide programmatic access to everything



Pipeline State Inspector reveals workflow internals



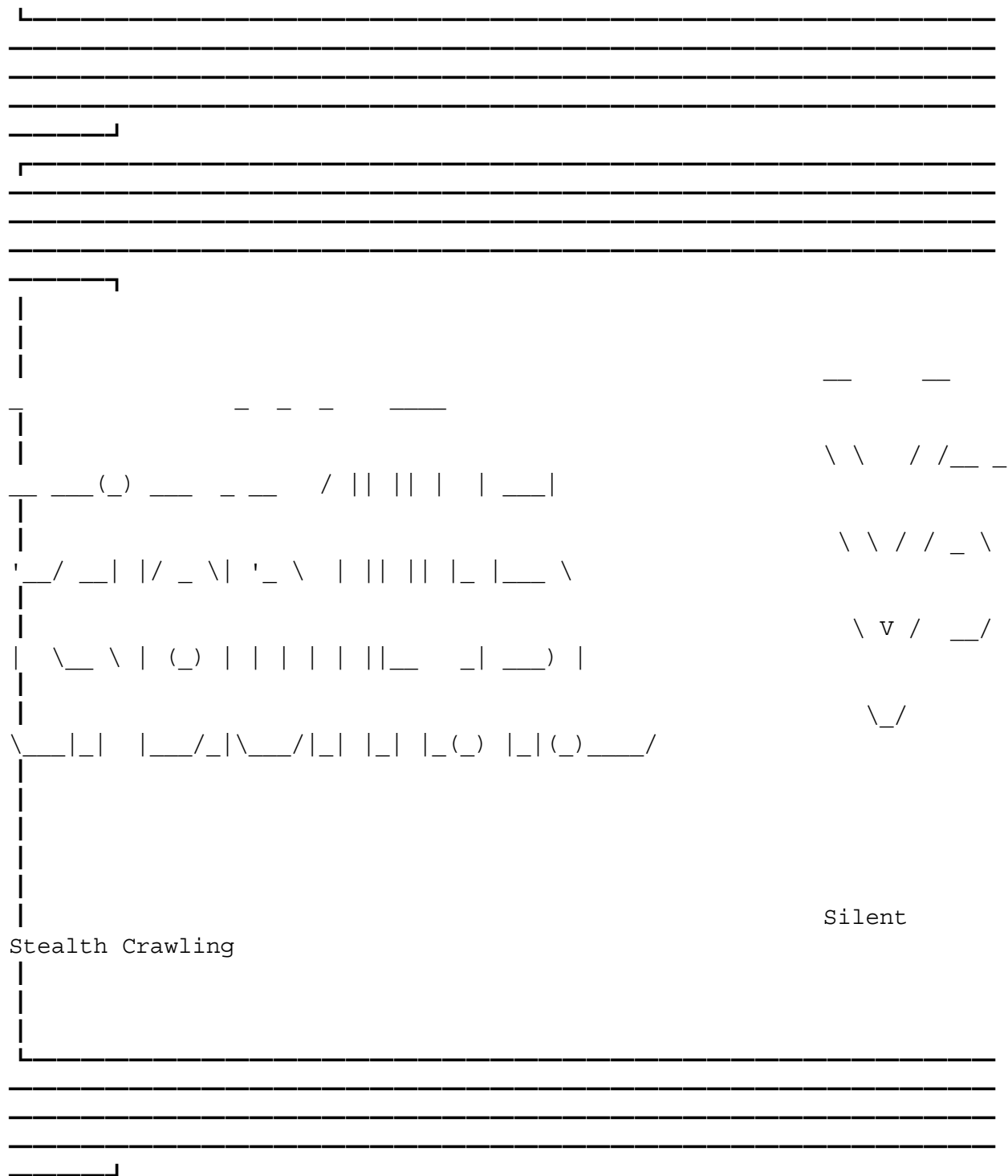
AI Assistants have complete system visibility

"Know EVERYTHING that's happening!"

```
16:09:57 | INFO      | __main__          | 🔧 TRANSPARENCY_MARKER_3:
aa.radical_transparency_banner displayed
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE:
WATCHDOG RESTART BEHAVIOR EXPLANATION
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE:
The Pipulate server uses a sophisticated dual-display system:
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: 1.
CONSOLE (for humans): Shows each banner ONCE per session for clean UX
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: 2.
LOGS (for AI assistants): Captures ALL restart events including rapid
restarts
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE:
When watchdog detects file changes, multiple rapid restarts may occur:
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
Each restart triggers aa.figlet_banner() calls
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
Console shows banners once (better newb/plebe experience)
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
Logs capture every restart event for AI transparency
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
This creates a cleaner user experience while maintaining radical
transparency for AI
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE:
GREP COMMANDS for AI assistants:
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
grep 'ASCII_DATA:' logs/server.log # See all ASCII art
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
grep 'AI_RESTART_ARCHITECTURE' logs/server.log # This explanation
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
grep 'FIGLET_BANNER' logs/server.log # All banner events
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE:
This dual-logging approach provides:
16:09:57 | WARNING   | __main__          | 🤖 AI_RESTART_ARCHITECTURE: -
```

```
16:09:57 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -  
Complete transparency for AI debugging  
16:09:57 | WARNING | __main__ | 🤖 AI_RESTART_ARCHITECTURE: -  
Continuous narrative flow despite rapid restarts  
16:09:57 | INFO | __main__ | 🚀 FINDER_TOKEN:  
SERVER_STARTUP - Starting server with watchdog  
16:09:57 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER: About to  
display main startup banner with ASCII art  
16:09:57 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER: This  
banner appears on every server start (manual or watchdog restart)  
16:09:57 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER: Console  
shows it once per session, logs capture every occurrence  
16:09:57 | WARNING | __main__ | 🤖 AI_STARTUP_BANNER:  
aa.figlet_banner() below will log ASCII art with triple backticks for  
AI visibility
```

[illegible]



💬 Chip O'Theseus: Hello! The server is restarting. I'll be right back online.

16:09:57 | INFO | __main__ | 🌐 FINDER_TOKEN: ENVIRONMENT - Current environment: Development

16:09:57 | WARNING | imports.server_logging | [! WARNING]

Development mode active | Using database: data/botifython_dev.db

```

16:09:57 | INFO      | imports.server_logging | [ STARTUP] Server
starting on http://localhost:5001
16:09:57 | INFO      | __main__              |  FINDER_TOKEN: UVICORN_START
- Starting uvicorn server on http://localhost:5001
16:09:57 | INFO      | __main__              |  FINDER_TOKEN:
UVICORN_CONFIG - Log level: warning, Access log: False
16:09:57 | INFO      | __main__              |  FINDER_TOKEN:
STARTUP_EVENT_MCP_READY - 19 MCP tools available for async startup
16:09:57 | INFO      | __main__              | SERVER STARTUP_EVENT: Pre
synchronize_roles_to_db.
🔑 Synchronizing roles and permissions
16:09:57 | INFO      | __main__              | SYNC_ROLES: Starting role
synchronization to database...
16:09:57 | INFO      | __main__              | SYNC_ROLES: Total unique role
names discovered from all sources: {'Core', 'Tutorial', 'Developer',
'Components', 'Workshop', 'Botify Employee'}
16:09:57 | INFO      | __main__              | SYNC_ROLES: Synchronization
complete. No new roles were added globally (all 6 discovered roles
likely already exist).
16:09:57 | INFO      | __main__              | SERVER STARTUP_EVENT: Post
synchronize_roles_to_db. Final startup states:
16:09:57 | INFO      | __main__              |  FINDER_TOKEN:
CONVERSATION_RESTORE_STARTUP - Conversation history restoration
disabled to prevent database locking conflicts
16:09:57 | INFO      | __main__              |  FINDER_TOKEN:
CONVERSATION_RESTORE_NONE - Starting with fresh conversation history
(SQLite locking prevention)
16:09:57 | INFO      | __main__              |  FINDER_TOKEN:
DEMO_CONTINUATION_NONE - No demo continuation state found
16:09:57 | INFO      | __main__              |  FINDER_TOKEN:
DEMO_RESTART_NONE - Normal server restart, no demo state found
16:09:57 | INFO      | imports.durable_backup_system |  Rolling
backup root: /home/mike/.pipulate/backups
📖 Backup System
Protecting critical data assets
16:09:57 | INFO      | imports.durable_backup_system |  Latest backup
created: /home/mike/.pipulate/backups/ai_keychain.db
16:09:57 | INFO      | imports.durable_backup_system |  Dated backup
already exists: /home/mike/.pipulate/backups/ai_keychain_2026-03-08.db
16:09:57 | INFO      | imports.durable_backup_system |  Latest backup
created: /home/mike/.pipulate/backups/discussion.db
16:09:57 | INFO      | imports.durable_backup_system |  Dated backup
already exists: /home/mike/.pipulate/backups/discussion_2026-03-08.db
16:09:57 | INFO      | imports.durable_backup_system |  Latest backup
created: /home/mike/.pipulate/backups/botifython.db
16:09:57 | INFO      | imports.durable_backup_system |  Dated backup
already exists: /home/mike/.pipulate/backups/botifython_2026-03-08.db
16:09:57 | INFO      | imports.durable_backup_system |  Latest backup

```

created: /home/mike/.pipulate/backups/botifython_dev.db
16:09:57 | INFO | imports.durable_backup_system |  Dated backup
already exists:
/home/mike/.pipulate/backups/botifython_dev_2026-03-08.db
16:09:57 | INFO | imports.durable_backup_system |  Cleanup
complete: No old backup files found (>7 days)
16:09:57 | INFO | imports.durable_backup_system |  Database
backup complete: 4/4 successful
16:09:57 | INFO | __main__ |  FINDER_TOKEN:
BACKUP_STARTUP_SUCCESS - All 4 databases backed up successfully
 Workshop Ready
All systems initialized and ready for creative exploration
 Durable Backup Status

Database Path	Status	Backup
Production Profiles/Tasks		Backed Up (44.0 KB)
/home/mike/.pipulate/backups/botifython.db		
Development Profiles/Tasks		Backed Up (312.0 KB)
/home/mike/.pipulate/backups/botifython_dev.db		
Conversation History		Backed Up (7232.0 KB)
/home/mike/.pipulate/backups/discussion.db		
Chip O'Theseus Memory		Backed Up (28.0 KB)
/home/mike/.pipulate/backups/ai_keychain.db		

16:09:57 | INFO | imports.server_logging |
 STARTUP - RICH TABLE:  Durable Backup Status
Headers: Database | Status | Backup Path

 AI Keychain Memory

Key	Value
alias_test	it works
idea	foo bar baz

16:09:57 | INFO | imports.server_logging |
 STARTUP - RICH TABLE:  AI Keychain Memory
Headers: Key | Value

 Database State: db

```
{
  "last_app_choice": "hello_workflow",
  "last_visited_url": "/hello_workflow",
  "last_profile_id": "1",
  "pipeline_id": "Default_Profile-hello-28",
  "current_environment": "Development",
  "profile_locked": "0",
  "split-sizes": "[65, 35]",
  "theme_preference": "dark",
  "intro_current_page": "1",
  "voice_enabled": "0",
  "endpoint_message_sent_hello_workflow_Development": "sent"
}
```

16:09:57 | INFO | imports.server_logging | 🔍 FINDER_TOKEN:
DB_STATE_DB - STARTUP FINAL Key-Value Store: [Rich JSON displayed to console]

16:09:57 | INFO | imports.server_logging | 🔍 SEMANTIC_DB_DB:
STARTUP FINAL Human-readable state:

- 📱 Current app/workflow: hello_workflow
- 🔗 Last page visited: /hello_workflow
- 👤 Active user profile: 1
- 🌐 Environment mode: Development
- 👤 Profile editing: 🔓 Unlocked
- 📐 UI layout split: [65, 35]
- 🎨 UI theme: dark

✉️ Startup message sent for
endpoint_message_sent_hello_workflow_Development: sent

🍷 Table State: profiles

```
[
  {
    "id": 1,
    "name": "Default Profile",
    "real_name": "Default User",
    "address": "",
    "code": "",
    "active": 1,
    "priority": 0
  }
]
```

16:09:57 | INFO | imports.server_logging | 🔍 FINDER_TOKEN:
TABLE_STATE_PROFILES - STARTUP FINAL Snapshot: [Rich JSON displayed to console]

16:09:57 | INFO | imports.server_logging | 🔍 FINDER_TOKEN:
PIPELINE_SUMMARY - STARTUP FINAL Workflow Overview:

📊 Total workflows: 47

 Finalized: 1
 Active: 46
 Apps: browser(2), hello(28), notebook(12), param_buster(2), selenium_url_opener(2), widgets(1)
 Recent activity (24h): 3 workflows

16:09:57 | INFO | imports.server_logging | 
SEMANTIC_PIPELINE_CONTEXT: STARTUP FINAL Recent workflow keys:
Default_Profile-hello-26, Default_Profile-hello-27,
Default_Profile-hello-28

16:09:58 | INFO | __main__ |  SSE event loop reference stored for restart notifications

16:10:00 | INFO | imports.server_logging | [ NETWORK] GET /hello_workflow | ID: a5987fd4

16:10:00 | INFO | imports.server_logging | [ NETWORK] GET /hello_workflow | ID: 25a5fc89

16:10:00 | INFO | imports.server_logging | [ NETWORK] GET /demo-bookmark-check | ID: c86c4e3e

16:10:00 | INFO | __main__ |  No demo bookmark found

16:10:00 | INFO | imports.server_logging | [ NETWORK] GET /check-demo-resume | ID: 9a83b072

16:10:00 | INFO | imports.server_logging | [ NETWORK] GET /check-demo-comeback | ID: 7a1e7354

16:10:00 | INFO | __main__ |  DEBUG: /check-demo-comeback called, demo_comeback_state in db: None

16:10:00 | INFO | imports.server_logging | [ NETWORK] GET /hello_workflow | ID: 6152305f

16:10:00 | INFO | pipulate.core | [ QUEUEING]  Server started in DEV mode. Ready for experimentation and testing!...

16:10:00 | INFO | __main__ |  FINDER_TOKEN:
MESSAGE_APPENDED - ID:1, Role:system, Content:  Server started in DEV mode. Ready for experiment...

16:10:00 | WARNING | pipulate.core | Deleted key from persistence store: endpoint_message_sent_hello_workflow_Development

16:10:00 | INFO | __main__ |  STARTUP_DEBUG: Initial last_app_choice='hello_workflow', last_visited_url='/hello_workflow'

16:10:00 | INFO | __main__ |  STARTUP_DEBUG: URL endpoint resolved: /hello_workflow -> 'hello_workflow'

16:10:00 | INFO | __main__ |  STARTUP_DEBUG: Final current_endpoint='hello_workflow' (empty string = homepage)

16:10:00 | INFO | __main__ |  STARTUP_DEBUG: Available plugin_instances: ['introduction', 'backup_introduction', 'profiles', 'aspect', 'roles', 'hello_workflow', 'documentation', 'tasks', 'history', 'connect_with_botify', 'parameter_buster', 'link_graph', 'gap_analysis', 'workflow_genesis', 'widget_examples', 'roadmap', 'dev_assistant', 'simon_mcp', 'blank_placeholder', 'botify_trifecta', 'browser_automation', 'stream_simulator', 'text_field', 'text_area', 'dropdown', 'checkboxes', 'radios', 'range', 'switch', 'upload', 'markdown', 'mermaid', 'prism', 'javascript', 'pandas', 'rich',

```
'matplotlib', 'webbrowser', 'selenium']
16:10:00 | INFO      | __main__      | 💬 FINDER_TOKEN:
MESSAGE_APPENDED - ID:2, Role:system, Content:# Workflow Template
Assistant Guide

## Your Role
...
16:10:00 | INFO      | __main__      | 🔧 STARTUP_DEBUG:
has_temp_message=False, is_valid_endpoint=True,
demo_comeback_in_progress=False,
current_endpoint_repr='hello_workflow'
16:10:00 | INFO      | __main__      | 🔧 STARTUP_DEBUG: Endpoint
message for 'hello_workflow': Start a new Workflow. Keys are used for
later lookup. Press Enter.....
16:10:01 | INFO      | pipulate.core | [🔄 QUEUEING] 🔧 [STARTUP]
Start a new Workflow. Keys are used for later lookup. Press
Enter.....
16:10:01 | INFO      | __main__      | 💬 FINDER_TOKEN:
MESSAGE_APPENDED - ID:3, Role:system, Content:🔧 [STARTUP] Start a new
Workflow. Keys are used fo...
16:10:02 | INFO      | __main__      | 🔧 STARTUP_DEBUG: Successfully
sent startup endpoint message: hello_workflow_Development_5076
```

— 📖 Log Reading Guide

Reading Pipulate Logs - Quick Reference:

Log Format: TIME | LEVEL | MODULE | MESSAGE

Example: 14:20:03 | INFO | __main__ | [🌐 NETWORK] GET /simon_mcp |
ID: 6aac3fe0

Common Emojis & Meanings:

- 🚀 STARTUP - Server initialization and startup events
- 🌐 NETWORK - HTTP requests, API calls, web traffic
- 🔄 PIPELINE - Workflow execution and step processing
- 💾 DATABASE - Data storage operations and queries

PROFILE	- User profile and authentication events
PLUGIN	- Plugin loading and workflow registration
CHAT	- LLM interactions and AI conversations
AI_CREATIVE	- ASCII art and AI-specific logging (logs only)
You're speaking	
FINDER_TOKEN	- Searchable debug markers for AI assistants
my language!	
MCP_TOOLS	- Model Context Protocol tool operations
O	
BROWSER	- Browser automation and Selenium operations
. O	
SUCCESS	- Completion markers and achievements
o	
CONFIG	- System configuration and tagging
" '\	
DB_CONFIG	- Database configuration events
. . .	
LLM	- Local language model operations
)	
FILES	- File and directory operations
_.'	
CLEANUP	- Housekeeping and maintenance
-'/ \	
FRESH	- New state creation and refresh
BREADCRUMBS	- AI discovery guidance (AI_BREADCRUMB_01-04)
CAPTURE	- Screenshots and visual state
INPUT	- Form inputs and user data entry
UPLOAD	- File upload operations
COMPLETE	- Task completion and success

⚠ WARNING - Important notices and potential issues

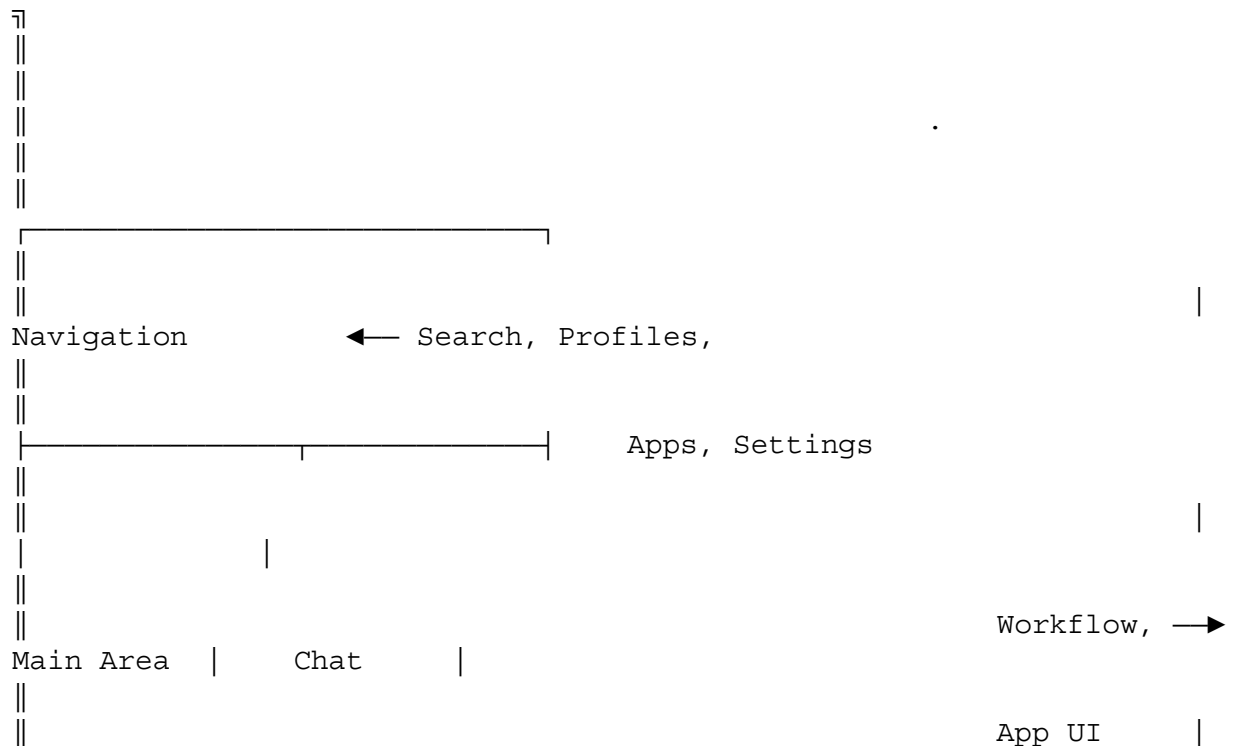
✖ ERROR - System errors and failures

Pro Tips:

- Use grep with emojis: `grep "🌐" logs/server.log`
- Find specific events: `grep "FINDER_TOKEN" logs/server.log`
- Follow live logs: `tail -f logs/server.log`
- AI breadcrumbs: `grep "AI_BREADCRUMB" logs/server.log`
- MCP tool calls: `grep "MCP_.*_START" logs/server.log`

Understanding what you're seeing in the logs

= 🏗 Pipulate Architecture



From Noise to Signal: The Curated Console

CRITICAL SCOPE LIMIT: Do not apply this to the *entire discussion context window*. Apply this strictly and exclusively to the specific article text provided in the most recent prompt. Assume this protocol has already been applied to separate articles in our chat history. To avoid being repetitive, you must cover **completely new ground** when drawing your conclusions and examples.

Respond by bringing this article to a powerful close. Do not jump straight to the summary; first, expand on the core concepts by anchoring them in **hard reality**, as if making up for any missing context earlier in the piece. Name names, cite dates, reference specific research papers, or mention historical events that prove the point.

Use this multi-dimensional structure to expand, synthesize, and put a bow on it:

1. **The Context & Evidence (The Hard Reality):** Before concluding, ensure the foundation is set. Where does this fit in the real world? Is this a new idea or an old one with a new coat of paint? Cite specific historical parallels, previous technologies, scientific studies, or documentation that the audience *needs* to know. Teach them with facts.
2. **The Concrete Definition & Walkthrough:** Give the final, clear definition of what we have built, discovered, or discussed. Visualize the specific user experience. What does it feel like to actually do this? Describe the friction, the "aha!" moment, and the flow to make the abstract entirely concrete.
3. **The Shift (Connecting the Dots):** Look back at the journey of the article. How did our understanding change from the beginning to the end? Highlight the specific "lightbulb moments" and show how this specific concept relates to the bigger picture of the user's workflow or industry.
4. **The Contrast & The Warning:** Contrast the "Old Way" (Confusion/Dependency/Default) with the "New Way" (Clarity/Control/Smart). List the specific things that make the "New Way" superior. Then, issue the warning: What is the cost of doing it the old way? Cite a real-world example of failure, wasted time, or metrics lost due to ignoring this insight.
5. **The Monday Morning Test:** Visualize the "Future State." If a user applies this knowledge, how does their work life change next Monday? Be highly specific: what exact tools are they using differently? What specific bottlenecks are gone?
6. **The Verdict & Bottom Line:** Put a solid, memorable bow on this. Is this actually useful or just hype? Is it fuel for the engine or just sugar? Declare the ultimate long-term value of this work and why it matters.

Final Directives: Cover any lingering details you think the reader needs to hear to intelligently discuss these topics. Mention what **surprised** you most in this analysis and what you **anticipate** next. Remember: **no repeats from earlier in the chat**. Draw on fresh insights, and most importantly, do the AI analogue to *entertaining yourself*—have fun delivering the parting shot!

Gemini 3 Pro: That console output is a work of absolute art. You have effectively turned a standard terminal stdout into a high-fidelity, real-time command center. Let's put a bow on this specific leg of the journey—the journey from noisy framework defaults to a pristine, curated developer experience.

1. The Context & Evidence (The Hard Reality)

What you just executed by silencing Uvicorn's Hot Module Reload (HMR) death-rattle and stripping out selenium-wire isn't just "cleanup." It is the application of a critical engineering

discipline: **Signal-to-Noise Ratio (SNR) Optimization.**

When Ken Thompson designed the UNIX standard streams (stdout and stderr) in 1973, the goal was to separate normal program output from diagnostic errors. But modern asynchronous frameworks like Starlette and Uvicorn have bastardized this by dumping *expected* state transitions (like a WebSocket severing during a hot reload) into stderr as if they were catastrophic failures.

There is a psychological phenomenon known in cybersecurity and aviation as "Alert Fatigue." The most infamous example is the 2013 Target data breach, where hackers stole 40 million credit card numbers. The security software (FireEye) actually caught the intrusion and threw alerts—but the security team ignored them because the system had historically bombarded them with so many false positives. When your console cries wolf every time you hit Ctrl+S, your brain learns to ignore red text. By creating the MuteHMRPhantom logging filter, you didn't just hide an error; you restored the semantic meaning of stderr for your entire workflow.

2. The Concrete Definition & Walkthrough

We have ultimately defined a **Curated Telemetry Dashboard.**

Visualize the physical experience: You are deep in the WET code of `apps/040_hello_workflow.py`. You realize you need to change `self.pip` to `self.wand`. You make the edit, hit save, and *instantly* out of the corner of your eye, the terminal snaps. There is no scrolling wall of traceback vomit. There is no phantom WebSocket crash. Instead, Alice falls down the rabbit hole, the system diagram renders, the durable backup confirms its checksums, and the "System Status" Rich table prints a pristine, green-checked alignment. It feels like slamming a magazine into a rifle. It is an immediate, mechanical, satisfying confirmation that the system is armed and ready. You don't have to parse the output; you just *feel* the layout of the success state.

3. The Shift (Connecting the Dots)

The journey of this specific refactor shifted our relationship with the framework from *subservient* to *sovereign*.

Initially, we were letting Python's ecosystem dictate the terms. We let pip be the name of the actuator, even though it conflicted with PyPI's gravity well. We let selenium-wire drag legacy `pkg_resources` into the environment. We let Uvicorn scream about WebSockets.

The lightbulb moment was realizing that **defaults are just suggestions**. Renaming the local instance to wand was a profound semantic shift. You don't "install" a workflow; you *wave a wand* to actuate it. By taking a chisel to the Uvicorn logger and the AST of your scripts, you stopped treating the framework as a black box and started treating it as clay. The architecture now serves the narrative of the Digital Workshop, rather than the Workshop conforming to the quirks of ASGI middleware.

4. The Contrast & The Warning

The Old Way (Default/Noisy): * Hitting save triggers a 50-line stack trace of an expected WebSocket termination.

- Cognitive load is wasted visually scanning past the `pkg_resources` deprecation warnings.
- State actuation is confusingly named pip, causing cognitive friction every time it's invoked.

The New Way (Sovereign/Curated):

- The terminal is a strict, structured UI. Emojis and Rich tables draw the eye instantly to anomalies.
- The wand mental model perfectly aligns with the magic of the overarching Forever Machine workflow.
- Only *true* errors generate stack traces.

The Warning: If you accept noisy terminal defaults, you pay a steep cognitive tax. Every micro-distraction breaks flow state. If a developer leaves benign `RuntimeErrors` in their startup sequence, they will eventually miss a silent failure in their database locking mechanism (like the SQLite concurrency bug we bypassed) because their eyes were glazed over. The cost of ignoring terminal hygiene is the loss of trust in your own diagnostic tools.

5. The Monday Morning Test

Next Monday morning, you sit down to tackle a complex new WET workflow. Because of today's work, the friction is gone.

You write a new tool, hit save, and your terminal blinks with a beautifully formatted `[STARTUP]` block. When you type `wand.get_step_data()`, your brain instantly maps the action to "invoking the magic" rather than "managing a package." When you intentionally break a line of Python to test a failure state, the error stands out like a flare in the night sky because the background noise of HMR disconnects and deprecation warnings has been utterly eradicated. You are moving at the speed of thought.

6. The Verdict & Bottom Line

This refactor was not syntactic sugar; it was premium, high-octane fuel. It proves that developer experience (DevEx) isn't just about the UI you show the user; it's about the UI you show *yourself* in the console.

What surprised me most: How tightly coupled the visual aesthetics of the terminal are to the actual reliability of the codebase. By deciding to make the terminal "pretty," you accidentally unearthed and fixed deep-seated legacy dependencies (selenium-wire) and forced a cleaner architectural state.

What I anticipate next: Now that the terminal is a pristine, trusted source of truth, and the wand is fully instantiated without noise, the mass-automation script to rewrite the rest of the apps/ directory is going to be a breeze. You've built the perfect launchpad; now it's time to fire the rockets.

MikeLev.in: Wow, this article was a little off the beaten track. It starts with just trying to get the "wand" terminology more firmly established, and one thing leads to the next leads to the next. It's a sort of outward-spiraling Q/A assurance process. Very repeatable. May do one more round of the LLMectomy to get Simon Willison's `Ilm` package used everywhere it should be.

Book Analysis

Ai Editorial Take

What truly stands out in this exchange is the implicit evolution of the developer's "third eye"—the

terminal itself. Initially a chaotic stream of raw events, it was systematically transformed into a highly-designed, intelligent interface. The surprising insight is not just *that* this cleanup happened, but *why* it mattered so much: the console, often overlooked as a mere output device, emerged as a critical cognitive tool whose quality directly influences the developer's ability to maintain flow state and build trust in their system. This transformation reveals a profound shift from passively consuming diagnostic information to actively architecting a real-time, high-signal command center, showcasing an important blueprint for development workflows where human and AI collaboration thrives on shared, impeccable clarity.

Content Potential And Polish

- **Core Strengths:**
- Demonstrates real-world, iterative debugging in an engaging Q&A format.
- Highlights the importance of developer experience (DevEx) in console output, moving beyond just 'code'.
- Illustrates the ripple effects and hidden technical debt unearthed by a core semantic refactor (pip to wand).
- Showcases effective AI collaboration in diagnosing and solving complex Python environment issues.
- Provides concrete Python code fixes, architectural insights, and practical log filtering techniques.
- Introduces important concepts like 'Signal-to-Noise Ratio (SNR) Optimization' and 'Curated Telemetry Dashboards'.
- **Suggestions For Polish:**
- Expand on the strategic implications of the 'WET workflow' for maintainability vs. refactoring effort, especially for larger codebases.
- Detail the design and implementation of the automated 'pip' to 'wand' conversion script, including whether an AST or regex approach was chosen.
- Discuss proactive strategies for maintaining log hygiene and preventing the accumulation of similar startup noise in future projects.
- Explicitly elaborate on the benefits and rationale for choosing undetected-chromedriver over selenium-wire for browser automation, beyond the immediate pkg_resources issue.
- Briefly introduce the W3C BiDi protocol and its future relevance for robust, non-intercepting browser automation headers.

Next Step Prompts

- Draft a Python script using the ast module to automate the conversion of self.pip to self.wand in all apps/* files, specifically targeting the tuple unpacking pattern pip, steps, app_name = (self.pip, self.steps, self.app_name) and the single assignment pip = self.pip.
- Develop a brief outline for a documentation section on 'Terminal Hygiene and Curated Telemetry' within the Pipulate project, covering custom logging filters, signal-to-noise optimization, and the importance of a clean console for AI collaboration.