

INSTALLATION & USAGE GUIDESYSTEM ONE HARNESS BY HARNESSROUTER

Turn a System One model (Jev) into an agent • One model call per step • Risk-based confidence gates • MCP, Browser, UHP

★ **GitHub Repository:** <https://github.com/HarnessRouter/SystemOneHarness> (~120+ Stars)

👑 **Tác giả / Tổ chức:** HarnessRouter

License: Apache-2.0

Language: Python 3.10 or newer (version 0.4.0)

Supported model: Jev by TypeSafe, via OpenRouter or TypeSafe directly

Key tech: httpx, PyYAML; optional MCP, Browser Use; UHP protocol

1. WHAT IS SYSTEM ONE HARNESS?

A System One model such as Jev does not generate text: it answers typed questions (choice – pick one of N, noul – yes/no, score – an ordered level) with probabilities, in a few hundred milliseconds. System One Harness is the control loop that turns such a model into an agent: it observes an environment, compiles the available actions into typed questions, calls the model exactly once per step, gates each decision by confidence before executing it, and records the complete trace.

- **Finite actions:** The model only chooses among actions and parameter values declared by the environment; it never writes commands.
- **Confidence gates:** Read, write, and destructive actions each have their own probability threshold (e.g. 0.5 / 0.6 / 0.8).
- **Explicit outcomes:** Every run ends as completed, incomplete, failed, or cancelled with a structured reason.
- **Complete traces:** State, questions, distributions, gate verdicts, latency, and cost are recorded step by step.
- **Pluggable environments:** Python processes, MCP servers, a browser (Browser Use), or an Environment subclass in Python.
- **UHP compatibility:** Serve the loop as a Unified Harness Protocol server; the README reports 40/40 checks in the core class.

2. SYSTEM REQUIREMENTS

- Python 3.10 or newer and pip.
- Git to clone the repository.
- An OpenRouter (OPENROUTER_API_KEY) or TypeSafe (TYPESAFE_API_KEY) API key to call Jev.

- Optional: the mcp package ([mcp] extra) for MCP servers; browser-use ([browser] extra) and Chrome for the browser environment.

3. STEP-BY-STEP INSTALLATION

Step 1: Clone and install

```
git clone https://github.com/HarnessRouter/SystemOneHarness.git
cd SystemOneHarness
pip install -e .
```

Step 2: Set your API key

```
# macOS / Linux
export OPENROUTER_API_KEY=sk-or-...    # hoặc / or: export
TYPESAFE_API_KEY=...
# Windows PowerShell
$env:OPENROUTER_API_KEY = "sk-or-..."
```

Step 3: Run the built-in example (order fulfilment environment)

```
sl run --env order:ship_fastest_gift
```

Expected result

Each line is one step: the selected action, its weakest required probability, and the model round trip. The final line shows the status, step count, wall time, and cost.

Sample output (from the README)

```
goal: Order B-220 is a gift: note it, then ship it by the fastest carrier.
model: ~typesafe/jev-latest
  0  add_note(note='gift')                p=0.96  241 ms
  1  pick_item(item='scarf')              p=1.00  269 ms
  2  pack()                              p=0.99  166 ms
  3  choose_carrier(carrier='express')    p=0.98  151 ms
  4  ship()                              p=0.93  152 ms

status=completed reason=environment_terminal steps=5 wall=0.98s cost=$0.000208
```

4. USAGE: CONNECT YOUR OWN ENVIRONMENT

Declare an action space in YAML: each action has a description, a risk level (read / write / destructive), and enumerable parameters (choices, observation candidates, flag, or levels). The gate section sets the probability threshold for each risk level.

actions.yaml (README example)

```
instructions: >-
  Move the order to shipped, or cancel it when the goal says so.
```

```

actions:
  choose_carrier:
    description: Select a carrier for the packed order.
    risk: write
    params:
      carrier:
        from: available_carriers
  ship:
    description: Hand the packed order to the selected carrier.
    risk: destructive

gate:
  read: 0.5
  write: 0.6
  destructive: 0.8
  finish: 0.5

```

Environment types

```

# Action space + tiến trình / process
s1 run --actions actions.yaml --env-cmd "python3 env.py" --goal "..."

# MCP server
pip install -e ".[mcp]"
s1 tools --mcp "python -m systemone_harness.envs.order_mcp"
s1 run --mcp "python -m systemone_harness.envs.order_mcp --scenario
ship_fastest_gift" \
  --goal "Order B-220 is a gift. Ship it by the fastest carrier."

# Browser (Browser Use)
pip install -e ".[browser]"
s1 run --browser --headless --start-url https://example.com/book \
  --text name=Customer --text email=user@example.com \
  --goal "Book a table at 19:30 with a window seat."

```

Serve over UHP

```

s
1
s
e
r
v
e
-
-
a
p

```

i - k e y c h o o s e - a - s e c r e t - - p o r t 8 7 1 0	
C o m m a n d	P u r p o s e
s 1 r u n	R u n o n e g o a l a

	g a i n s t a b i l i t - i n , p r o c e s s , M C P , o r b r o w s e r e n v i r o n m e n t
s 1	I n

t o o l s	s p e c t h o w a n M C P s e r v e r c o m p i l e s i n t o s u p p o r t e d a c t i o n s
s 1 s	E x p

e r v e	o s e a c o n f i g u r e d l o o p a s a U H P s e r v e r
s 1 b e n c h	R u n t h e b u i l t - i n l i v e b e

	n
	c
	h
	m
	a
	r
	k

5. COMMON ERRORS & FIXES

- **Missing API key:** Set `OPENROUTER_API_KEY` or `TYPESAFE_API_KEY` in the same terminal before running `s1 run`.
- **s1 command not found:** Re-run `pip install -e .` inside the repo and make sure your Python Scripts/bin directory is on PATH (or use a virtualenv).
- **Free-text parameter rejected:** A System One model does not generate text, so every parameter must be enumerable (choices, candidates, flag, levels). For the browser, pass values with `--text name=value`.
- **MCP tool skipped:** Run `s1 tools --mcp "..."` to see why each tool is unsupported; install the `[mcp]` extra if the package is missing.
- **Run stops with no_confident_action / escalate:** No action cleared its gate. Inspect the handoff in the trace (`--json trace.json`) for the refused questions and probabilities, then clarify the observation or tune thresholds.
- **Browser environment import error:** Install the extra: `pip install -e ".[browser]"`. See `docs/browser-use.md` for limits and measurements.

Run the test suite (no key needed)

```
pip install -e . pytest
pytest -q tests
```

Limits stated by the authors

The README benchmark only proves small deterministic tasks. It makes no claim for ambiguous state, arithmetic, dates, or long irrelevant context.

Build fast, cheap, and transparent reflex agents with System One Harness!

© Kênh AI Mỗi Ngày | Hướng dẫn AI & Linux Tự Động Hóa

Nội dung kỹ thuật tổng hợp từ tài liệu chính thức của repo <https://github.com/HarnessRouter/SystemOneHarness>