

CẨM NANG CÀI ĐẶT & SỬ DỤNG SYSTEM ONE HARNESS BY HARNESSROUTER

Biến model System One (Jev) thành agent • 1 lần gọi model mỗi bước • Confidence gate theo rủi ro • MCP, Browser, UHP

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

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

Giấy phép: Apache-2.0

Ngôn ngữ: Python 3.10 trở lên (phiên bản 0.4.0)

Model hỗ trợ: Jev của TypeSafe, qua OpenRouter hoặc TypeSafe trực tiếp

Công nghệ chính: httpx, PyYAML; tùy chọn MCP, Browser Use; giao thức UHP

1. SYSTEM ONE HARNESS LÀ GÌ?

Model System One như Jev không sinh text: nó chỉ trả lời câu hỏi có kiểu (choice – chọn 1 trong N, noul – có/không, score – thang bậc) kèm xác suất, trong vài trăm mili giây. System One Harness là vòng lặp điều khiển biến model đó thành agent: quan sát môi trường, biên dịch các action khả dụng thành câu hỏi có kiểu, gọi model đúng một lần mỗi bước, kiểm tra độ tự tin rồi mới thực thi, và ghi lại toàn bộ trace.

- **Action hữu hạn:** Model chỉ chọn trong các action và giá trị tham số do môi trường khai báo, không bao giờ tự viết lệnh.
- **Confidence gate:** Action đọc, ghi và phá hủy có ngưỡng xác suất riêng (ví dụ 0.5 / 0.6 / 0.8).
- **Kết thúc rõ ràng:** Mỗi lượt chạy kết thúc là completed, incomplete, failed hoặc cancelled, kèm lý do có cấu trúc.
- **Trace đầy đủ:** Lưu trạng thái, câu hỏi, phân phối xác suất, kết quả gate, độ trễ và chi phí từng bước.
- **Môi trường cấm được:** Tiến trình Python, MCP server, trình duyệt (Browser Use) hoặc lớp Environment trong Python.
- **Tương thích UHP:** Chạy như server Unified Harness Protocol; README báo cáo đạt 40/40 kiểm tra lớp core.

2. YÊU CẦU HỆ THỐNG

- Python 3.10 trở lên và pip.
- Git để clone repo.
- API key OpenRouter (OPENROUTER_API_KEY) hoặc TypeSafe (TYPESAFE_API_KEY) để gọi Jev.
- Tùy chọn: gói mcp (extra [mcp]) cho MCP server; browser-use (extra [browser]) và Chrome cho môi trường trình duyệt.

3. CÀI ĐẶT TỪNG BƯỚC

Bước 1: Clone repo và cài đặt

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

Bước 2: Đặt API key

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

Bước 3: Chạy ví dụ có sẵn (môi trường xử lý đơn hàng)

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

Kết quả mong đợi

Mỗi dòng là một bước: action được chọn, xác suất yếu nhất và thời gian gọi model.
Dòng cuối cho biết trạng thái, số bước, tổng thời gian và chi phí.

Output mẫu (từ 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. SỬ DỤNG: KẾT NỐI MÔI TRƯỜNG CỦA BẠN

Khai báo action space bằng YAML: mỗi action có mô tả, mức rủi ro (read / write / destructive) và tham số liệt kê được (choices, candidates từ quan sát, flag hoặc levels). Phần gate đặt ngưỡng xác suất cho từng mức rủi ro.

actions.yaml (ví dụ trong README)

```
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

```

Các loại môi trường

```

# 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."

```

Chạy như server UHP

```

s1 serve --api-key
choose-a-secret
--port 8710

```

Lệnh	Chức năng
s1 run	Chạy một goal với môi trường built-in, process, MCP hoặc browser
s1 tools	Xem MCP server được biên dịch thành những action nào
s1 serve	Mở vòng lặp dưới dạng server UHP

s1 bench	Chạy benchmark live có sẵn
----------	----------------------------

5. LỖI THƯỜNG GẶP & CÁCH KHẮC PHỤC

- **Thiếu API key:** Đặt OPENROUTER_API_KEY hoặc TYPESAFE_API_KEY trong cùng terminal trước khi chạy s1 run.
- **Không tìm thấy lệnh s1:** Chạy lại pip install -e . trong thư mục repo và đảm bảo thư mục Scripts/bin của Python nằm trong PATH (hoặc dùng virtualenv).
- **Tham số dạng free text bị từ chối:** Model System One không sinh text, nên mọi tham số phải liệt kê được (choices, candidates, flag, levels). Với trình duyệt, truyền giá trị qua --text name=value.
- **Tool MCP bị bỏ qua:** Dùng s1 tools --mcp "..." để xem lý do từng tool không được hỗ trợ; cài extra [mcp] nếu thiếu gói.
- **Run dừng với no_confident_action / escalate:** Không action nào vượt ngưỡng gate. Xem mục handoff trong trace (--json trace.json) để biết câu hỏi và xác suất bị từ chối, rồi làm rõ quan sát hoặc điều chỉnh ngưỡng.
- **Môi trường trình duyệt lỗi import:** Cài extra: pip install -e ".[browser]". Xem docs/browser-use.md để biết giới hạn và cách đo.

Chạy test suite (không cần key)

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

Giới hạn được tác giả nêu rõ

Benchmark trong README chỉ chứng minh các tác vụ nhỏ, xác định. Kết quả chưa được khẳng định cho trạng thái mơ hồ, số học, ngày tháng hay ngữ cảnh dài không liên quan.

Tài khoản githot mới được tặng 5 xu để dùng thử công cụ AI Repo Explainer hoặc Repo Health Check.

Đăng ký: githot.io.vn/register · Tìm thêm hướng dẫn cài đặt: githot.io.vn

Chúc bạn xây dựng những agent phản xạ nhanh, rẻ và minh bạch cùng 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>