Trang 5.8: Quản lý môi trường xem trước và theo dõi tiến độ (/preview & /status).

1. Tổng quan

Trang này hướng dẫn hai workflow đơn giản nhưng cực kỳ hữu ích trong công việc hàng ngày:

Workflow	Mục đích
`/preview`	Quản lý server phát triển cục bộ
`/status`	Xem tổng quan tiến độ và trạng thái dự án

Phần A: `/preview` — Quản lý server phát triển

A.1. Các lệnh `/preview`

Lệnh	Chức năng
`/preview start`	Khởi động server phát triển
`/preview stop`	Dừng server đang chạy
`/preview restart`	Khởi động lại server (áp dụng config mới)

Lệnh	Chức năng
`/preview`	Kiểm tra trạng thái hiện tại

A.2. Tính năng tự động nhận diện framework

Workflow này đủ thông minh để **tự nhận diện** loại dự án và chạy đúng lệnh:

Framework	Lệnh được chạy tự động
Next.js	`npm run dev` (port 3000)
Vite	`npm run dev` (port 5173)
Create React App	`npm start` (port 3000)
Express/Node.js	`node server.js` hoặc `npm start`
FastAPI (Python)	`uvicorn main:app --reload`

A.3. Xử lý xung đột Port

Nếu port mặc định (3000) đã bị chiếm bởi ứng dụng khác:

```
AI: Port 3000 đang bận.
Đề xuất alternatives:
- Port 3001  (available)
- Port 8080  (available)


Dùng port 3001? (Y/N)
```

A.4. Ví dụ phiên làm việc

```
Bạn: /preview start


AI:
  Đang phát hiện framework...
  Tìm thấy: Next.js 14 (App Router)
```

□□ Đang khởi động...

Running: npm run dev

□ Preview sẵn sàng!

URL: http://localhost:3000

Server sẽ tiếp tục chạy. Dùng /preview stop để tắt.

□□ Bạn: /preview

□□ AI:

□□ Preview Status

Status: □□ Running

URL: http://localhost:3000

Framework: Next.js 14

Port: 3000

Uptime: 12 phút

□□ Bạn: /preview stop

□□ AI:

□ Đã dừng server preview.

Port 3000 đã được giải phóng.

A.5. Khi nào dùng `/preview`

Tình huống	Lệnh
Sau khi `/create` hoặc `/enhance` xong	`/preview start`
Cần xem UI thay đổi trên trình duyệt	`/preview start`

Tình huống	Lệnh
Thay đổi cấu hình (next.config.js...)	`/preview restart`
Trước khi tắt máy	`/preview stop`
Không nhớ server có đang chạy không	`/preview`

Phần B: `/status` — Bảng điều khiển tiến độ dự án

B.1. Mục đích

Lệnh `/status` cung cấp **bảng điều khiển tổng thể** (dashboard) để bạn nắm bắt nhanh tình trạng hiện tại của dự án, bao gồm: thông tin kỹ thuật, tiến độ các Agent đang làm việc, và trạng thái server.

B.2. Cú pháp

```
bash
/status
```

B.3. Thông tin được hiển thị

```
=== Project Status ===



□□ Project Information
Name: my-ecommerce-app
Type: Next.js + PostgreSQL
Version: 1.2.0
Files: 147 total (73 created, 12 modified, 62 unchanged)



=== Agent Status ===
```

 database-architect → Completed (Schema + Migrations)

 backend-specialist → Completed (All API Endpoints)

 frontend-specialist → In Progress (60% — Cart Page)

 test-engineer → Waiting (Pending frontend completion)

=== Test Results ===

 Unit Tests: 45/45 passed

 Integration Tests: 12/12 passed

⚠ E2E Tests: 8/10 passed (2 failing)

=== Preview ===

 URL: http://localhost:3000

 Health: OK (responding in 124ms)

B.4. Ví dụ phiên làm việc

 Bạn: /status

 AI:

=== PROJECT STATUS ===

 my-saas-app (Next.js + Prisma + PostgreSQL)

Files: 89 tổng (34 mới tạo hôm nay)

Branch: feature/payment-integration

=== □□ AGENT STATUS ===

□ database-architect → DONE — Payment schema + Stripe webhook table
□ backend-specialist → DONE — /api/payments (create, confirm, refund)
□ security-auditor → DONE — Reviewed, không có high-severity issues
□□ frontend-specialist → 70% — Đang tạo CheckoutPage...
□ test-engineer → Chưa bắt đầu

=== □□ TEST STATUS ===

□ 38 tests passed | □ 0 failed | □□ Coverage: 82%

=== □□ PREVIEW ===

Status: □□ Running — http://localhost:3000
Response time: 89ms

=== □□ NEXT STEPS ===

1. Chờ frontend-specialist hoàn thành CheckoutPage (~30 phút)
2. Chạy /test để tạo test cho payment flow
3. Chạy /deploy khi tất cả test pass

B.5. Khi nào dùng `/status`

Tình huống	Lý do
Đầu ngày làm việc	Nhắc lại tiến độ từ hôm qua
Sau khi `/orchestrate` phức tạp	Kiểm tra tất cả Agent đã xong chưa
Trước khi `/deploy`	Đảm bảo mọi thứ xanh trước khi đẩy lên production
Khi bàn giao cho đồng đội	Chia sẻ trạng thái hiện tại nhanh chóng
Khi không nhớ đang làm đến đâu	Lấy lại ngữ cảnh công việc

Phần C: Vòng lặp phát triển kết hợp

Sử dụng `/preview` và `/status` trong quy trình hàng ngày:

□□ Bắt đầu ngày làm việc:

/status → Xem lại tiến độ hôm qua

/preview start → Khởi động server để làm việc

□□ Trong giờ làm:

/enhance [feature] → Thêm tính năng

/preview start → Xem kết quả

/test → Kiểm tra không có regression

□□ Cuối ngày:

/status → Tổng kết tiến độ hôm nay

/preview stop → Tắt server

□□ Trước khi deploy:

/status → Xác nhận tất cả Agent xong việc

/test → Đảm bảo tất cả test pass

/deploy → Triển khai

Phiên bản #2
Được tạo 2026-03-04 06:12:39 UTC bởi Nam Đặng
Được cập nhật 2026-03-04 09:59:39 UTC bởi Nam Đặng