

Inyección Headless en WooCommerce con Python Puro

Cuadernillo | Vol. 1

mercedev.es — 2026-04-30 | Fase 8 (Expansión de Contenido)

El Desafío (Síntoma)

Para auditar el diseño SASS de una tienda mínima viable (MVP) en el entorno de desarrollo local, era imperativo disponer de productos de prueba. Acceder al panel de administración de WordPress (GUI) para crear datos falsos genera fricción operativa y rompe el paradigma "CLI-first" (Interfaz de Línea de Comandos primero) de nuestra arquitectura.

La Maniobra (Lógica)

Se desarrolló un script de un solo uso (`merci-wc-mock.py`) utilizando exclusivamente la librería estándar de Python (`urllib.request`). El script consume de forma segura las credenciales almacenadas en el archivo local `.env` , las codifica en Base64 (Basic Auth) y dispara una petición POST con un *payload* en formato JSON directamente contra el endpoint nativo de WooCommerce (`/wp-json/wc/v3/products`).

El Aprendizaje / Deuda Técnica

Este hito demuestra la verdadera potencia de una arquitectura Headless acoplada a un CMS monolítico. WooCommerce no es solo un plugin de comercio electrónico, es un motor de base de datos con una API REST completa.

Al abstenernos de usar comandos `curl` crudos (que filtrarían las contraseñas en el historial bash) y rechazar librerías externas de terceros (como la dependencia de WP para Python), demostramos que la automatización Shift-Left y la inyección de datos pueden realizarse de forma ultrasegura y con latencia cero empleando las herramientas base del sistema operativo.