

Crypto Token-Gated Download Template for Digital Products (KO)

[ko]\n\n# Crypto Token-Gated Download Template for Digital Products — Practical Digital Product Guide (KO)

Who this is for

- People who accept payments in crypto and want a clean, reusable checkout experience
- Solo builders who need templates, checklists, and ready-to-ship assets
- Anyone who wants a repeatable “publish → promote → sell” loop

What you get

- A ready-to-ship landing + checkout flow (mock-safe)
- Token-gated download concept (server-side verification)
- Product packaging & bonus assets
- Promotion copy for major channels

Quick Start Checklist

- [] Generate product package (auto_pilot)
- [] Run dashboard (dashboard_server.py)
- [] Run payment mock server (backend/payment_server.py)
- [] Confirm /api/pay/start → /api/pay/check → /api/pay/token → /api/pay/download

Architecture Overview

Components

- 1) Product Generator
- 2) Quality Control (QC)
- 3) PDF Builder
- 4) Bonus Pack Builder
- 5) Promotion Generator
- 6) Dashboard + Scheduler
- 7) Payment + Token-gated Downloads

Token-gated download (concept)

- Server issues token after paid status
- Token contains order_id, product_id, exp
- Server verifies token signature + expiry
- Server re-checks paid status (defense in depth)

Implementation Notes

Mock mode behavior

- No API keys → system still runs fully
- Payment becomes "admin mark paid" test flow
- Translation becomes mock translation

Security notes (practical)

- Use strong PAYMENT_TOKEN_SECRET in production
- Keep TTL short (e.g., 10 minutes)
- Do not trust client status; re-check server store

Bonus Assets (included)

- Checklists
- Prompt library
- Release steps

- Mini FAQ

Promotion Copy Templates

Blog (long form)

- Problem → Solution → Proof → CTA

Instagram / TikTok / Shorts (short form)

- Hook → Value → Proof → CTA

Appendix

Troubleshooting

- If ports conflict: change DASHBOARD_PORT / PAYMENT_PORT

- If PDF fails: ensure reportlab installed