

SW development

Skripty od ChatGPT (či jiné AI)

Jak zadávat požadavky ChatGPT

- Dát modelu **specifikaci prostředí**: OS, verze Pythonu/Node, dostupné knihovny, omezení (proxy, firewall).
- Požadovat **strukturovaný výstup** (JSON/patch/diff), aby se dal snadno aplikovat.
- Žádat **nejmenší nutnou změnu** („minimal patch“), ne vždy celý soubor.
- Nechat si vypsát **edge cases** a generovat k nim testy.
- Požadovat **komentáře k designovým volbám** (proč X, ne Y).

Spouštění & konfigurace

- **Konfigurační vrstva**: .env / YAML / CLI přepínače; precedence (ENV > CLI > config).
- **Verbose / dry-run / debug** módy.
- **Exit kódy** a jednotná chybová hlášení pro skriptování v CI.

Logování & pozorovatelnost (navázat na tvé body)

- **Úrovně logů** (INFO/DEBUG/WARN/ERROR) + **strukturované logy** (JSON).
- **Rotace logů** (size/time-based), ne jen „promazat při startu“.
- **Notifikace**: při pádu/úspěchu (e-mail, webhook, systémové upozornění), nejen „beep“.
- **Měření doby běhu**, počty zpracovaných položek, jednoduché metriky.

Spolehlivost & odolnost

- **Timeouty** na IO/HTTP a **retry** s exponenciálním backoffem + jitter.
- **Idempotence** (bezpečné opakování po pádu).
- **Dočasné soubory** a **atomické zápisy** (write-temp → rename).
- **Zámky**/file locking u souběžných běhů.
- **Graceful shutdown** (SIGINT/SIGTERM).

Bezpečnost

- **Správa tajemství** (API klíče jen z ENV/secret manageru; nikdy do repo).
- **Sanitizace vstupů**; pozor na **shell injection** u subprocess.
- **Omezení přístupů** (práce jen v určeném adresáři, práva souborů).
- **Redakce logů** (nevypisovat PII/klíče).

Testování & kvalita

- Nechat ChatGPT vygenerovat **unit testy** (pytest/unittest) a **falešná data**.
- **Lint/format** (ruff/flake8, black, prettier) + **type hints** (mypy/pyright).
- **CI kontrola**: testy, lint, bezpečnostní skeny (např. bandit).

Závislosti & prostředí

- **Reproducibilita**: requirements.txt/poetry.lock, pipx pro CLI, verze nástrojů.
- **Virtuální prostředí**/containers (Docker) pro izolaci.
- **Kompatibilita napříč OS** (cesty, kódování, CRLF/LF, lokály).

Vstup/Výstup & data

- **Kódování** (UTF-8), **normalizace konců řádků**.
- **Validace** vstupních souborů (CSV/JSON schéma).
- **Streamování** velkých dat, **batching**, **caching** (např. HTTP etag/If-Modified-Since).
- **Progres** (tqdm) u dlouhých běhů.

SW development

Paralelizace & výkon

- Volba mezi **asyncio** / **threads** / **multiprocessing** (IO vs CPU bound).
- **Omezení souběhu** (semaforey, rate-limit), aby se nezahltily API/disky.
- **Profilace** (time/perf counter) a základní optimalizace paměti.

Integrace ChatGPT API (pokud používáš)

- **Volba modelu** a **determinismus** (seed), **temperature** pro stabilnější kód.
- **JSON/structured output** a **tools/function calling** (schémata, Pydantic).
- **Kontrola nákladů**: logování tokenů, metriky, rozumné promptování.
- **Fallback** modelů, **rate-limit** handling, **streaming** vs. celé odpovědi.
- **Chunking** delších souborů, kontextová okna, **cache** odpovědí.

Sít & provoz

- **Proxy podpora** (HTTP/HTTPS), timeouts, retry politiky.
- **Plánování**: cron/Systemd/Task Scheduler, watchdog, restart po chybě.
- **Zdravotní kontrola** (health-check) pro dlouho běžící služby.

Dokumentace & údržba

- Krátké **README**: jak spustit, požadavky, příklady.
- **Verzování** skriptu a **changelog**.
- **Licence**/poznámky ke třetím knihovnám.

UX drobnosti

- **Barevný/čistý CLI výstup** (ale s přepínačem -no-color).
- **Interaktivní potvrzení** u nevratných operací, -yes pro automatizaci.
- **Sensible defaults** + jasná nápověda -help.

Co si nechat od ChatGPT rovnou vygenerovat

- **Skeleton projektu** (struktura adresářů).
- **Konfig šablony** (.env.example, YAML).
- **Makefile/Taskfile** pro běžné úlohy (test, lint, run).
- **CI workflow** (GitHub Actions/GitLab CI).
- **Migrační plán**/refactor diff a **rollback** instrukce.

Unique solution ID: #1175

Author: n/a

Last update: 2025-09-17 15:15