FICHE MÉMO DÉVELOPPEUR CLI, GIT & ARCHITECTURE

Outils Open Source, Raccourcis Terminal & Architecture Système (2026)

1. Flux de Travail Git Moderne & Contrôle de Version

Moteur VCS

- `git switch -c <branche>` — Établir une nouvelle branche et basculer aussitôt.
- `git commit --amend --no-edit` — Intégrer les fichiers oubliés au dernier commit.
- `git rebase -i HEAD~3` — Rebase interactif pour fusionner (squash) les commits.
- `git cherry-pick <hash>` — Rapatrier un commit précis depuis une autre branche.
- `git stash push -m "note"` — Mettre de côté le travail en cours; réappliquer via `pop`.
- `git restore --staged <fichier>` — Retirer un fichier de l'index sans perdre l'édition.
- `git reset --soft HEAD~1` — Annuler le commit tout en conservant les fichiers indexés.
- `git log --oneline --graph --all` — Visualiser l'arbre complet en ASCII clair.

2. GitHub CLI (gh) & Raccourcis Lazygit

TUI & CLI

Outil	Commande / Touche	Fonctionnalité & Gain de Productivité
gh	`gh pr create --web`	Ouvrir le navigateur pour soumettre une Pull Request.
gh	`gh pr checkout <no>`	Télécharger la branche PR en local et basculer dessus.
gh	`gh run watch`	Suivre en direct les flux d'intégration continue Actions.
lazygit	`Espace` / `c` / `P`	Indexer le fichier / Valider le commit / Pousser upstream.
lazygit	`z` / `e` / `[ ]`	Annuler l'action / Ouvrir dans l'éditeur / Changer d'onglet.

3. Gestionnaires de Paquets & Environnements Ultra-Rapides

uv • pnpm • fnm

```
# uv (Python - Conçu en Rust, 10 à 100x plus rapide que pip)
uv venv && source .venv/bin/activate # Environnement virtuel instantané
uv pip install -r req.txt # Cache global ultra-rapide
uv run script.py # Exécution dans un environnement isolé éphémère
```

```
# pnpm (Node.js - Économise 80% de disque via des liens durs)
pnpm install && pnpm add -D typescript # Dépôt adressable par le contenu
pnpm dlx degit <repo> <cible> # Exécution directe sans installation préalable
```

```
# fnm (Fast Node Manager - Bascule Node instantanée en Rust)
fnm install --lts && fnm use 22 # Zéro latence au démarrage du shell
```

4. Navigation Terminal & Raccourcis Readline

Raccourcis Clavier

Touche	Action	Touche	Action
`Ctrl + A`	Aller au début de la ligne	`Ctrl + E`	Aller à la fin de la ligne
`Ctrl + U`	Couper du curseur au début	`Ctrl + K`	Couper du curseur à la fin
`Ctrl + W`	Supprimer le mot précédent	`Ctrl + Y`	Coller le dernier texte coupé
`Ctrl + R`	Recherche inverse historique	`Ctrl + L`	Effacer l'affichage écran
`Alt + .`	Insérer le dernier argument	`Ctrl + C`	Interrompre le processus actif

5. Codes de Statut HTTP & Architecture REST

Standards API

Code	Signification	Usage Pratique & Règle d'Architecture
200 OK	Succès	Requête GET/PUT/PATCH traitée avec succès.
201 Created	Ressource Produite	POST a produit une ressource (En-tête Location retourné).
204 No Content	Corps Vide	Opération réussie; aucun corps retourné en réponse.
304 Not Mod.	Cache Valide	ETag vérifié; zéro octet de payload transmis.
400 Bad Req.	Requête Invalide	Syntaxe erronée ou schéma JSON rejeté.
401 Unauth.	Non authentifié	Identifiants manquants ou jeton d'accès expiré.
403 Forbidden	Accès Interdit	Identité vérifiée, mais droits RBAC insuffisants.
404 Not Found	Introuvable	La ressource demandée n'existe pas sur le serveur.
409 Conflict	Conflit d'État	Contrainte d'unicité violée ou collision de version.
422 Unproc.	Entité Rejetée	JSON syntaxiquement valide, mais validation métier échouée.
429 Rate Limit	Quota Dépassé	Trop de requêtes émises; examiner l'en-tête Retry-After.
500 / 502 / 503	Erreurs Serveur	Crash interne / Passerelle défaillante / Surcharge temporaire.

6. Expressions Régulières (Regex) Essentielles

Validation

E-Mail: `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`

URL (HTTP/S): `^https?:\/\/[\w.-]+(?:\.[\w\.-]+)+[\w\-\._~:/?#[\]@!\$&'\(\)\*\+,;=.]+$`

Adresse IPv4: `^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`

UUID v4: `^[0-9a-f]{8}-[0-9a-f]{4}-4[0-9a-f]{3}-[89ab][0-9a-f]{3}-[0-9a-f]{12}$`

Date ISO 8601: `^\d{4}-(?:0[1-9]|1[0-2])-(?:0[1-9]|[12]\d|3[01])$`

Version Sémantique: `^v?(0|[1-9]\d*)\.(0|[1-9]\d*)\.(0|[1-9]\d*)(?:-[0-9A-Za-z.-]+)?$`

7. Indexation SQL & Architecture Relationnelle

Règles SGBD

Règle du Préfixe Gauche: `INDEX(a, b, c)` accélère uniquement les requêtes sur `(a)`, `(a, b)` ou `(a, b, c)`. Une clause `WHERE b = 5` seule provoque un scan complet de la table.

Mécanique des JOIN: `INNER` produit l'intersection. `LEFT JOIN` conserve toutes les lignes de gauche. Éviter `SELECT *` pour limiter les surcoûts d'I/O.

Garanties ACID: Atomicité (tout ou rien), Cohérence (intégrité des règles), Isolation (étanchéité des transactions), Durabilité (persistance disque).

Analyse de Requête: Valider les requêtes en production avec `EXPLAIN ANALYZE` pour vérifier l'utilisation effective des index.

8. Outils CLI Modernes (Écosystème Rust & Go)

Accélérateurs

Outil Ancien	Outil Moderne	Avantage de Performance & Fonctionnalité
`cd`	**zoxide (z)**	Mémorise vos habitudes; saut direct via `z projet`.
`cat`	**bat**	Coloration syntaxique, numéros de ligne et intégration Git.
`grep`	**ripgrep (rg)**	Écrit en Rust, respecte le .gitignore, recherche foudroyante.
`find`	**fd**	Syntaxe claire avec parcours parallèle du système de fichiers.
`ls`	**eza**	Icônes, droits d'accès détaillés et statut Git en couleur.