

Git_Flow_Cheat_Sheet

Section	Command / Info	Notes
Branch Roles	main	Stable production code
Branch Roles	develop	Integration of new work
Branch Roles	feature/*	Feature development
Branch Roles	release/*	Release prep
Branch Roles	hotfix/*	Urgent production fixes
Start Feature	git checkout develop git pull origin develop git checkout -b feature/your-feature	Branch off from develop
Commit & Push	git add . git commit -m "Add feature XYZ" git push -u origin feature/your-feature	Stage, commit, and push with upstream
Merge Feature	git checkout develop git pull origin develop git merge feature/your-feature git push origin develop	Finish feature via merge to develop
Start Hotfix	git checkout main git pull origin main git checkout -b hotfix/fix-name	Start from main
Finish Hotfix	git add . git commit -m "Hotfix for production issue" git checkout main git merge hotfix/fix-name git push origin main git checkout develop git merge hotfix/fix-name git push origin develop	Merge hotfix into main and develop
Create Release	git checkout develop git checkout -b release/v1.0.0	Create release from develop
Finalize Release	git checkout main git merge release/v1.0.0 git tag -a v1.0.0 -m "Release version 1.0.0" git push origin main --tags git checkout develop git merge release/v1.0.0 git push origin develop	Merge release to main & tag version
Override your local branch	git fetch origin git reset --hard origin/main git clean -fd	Fetch the latest remote info Reset your local branch to match the Remote branch Remove untracked files (e.g., new files not committed)
Utilities	git branch -a	List all branches
Utilities	git branch -d feature/your-feature git push origin --delete feature/your-feature	Delete branches locally and remotely
Utilities	git stash git stash pop	Save and restore uncommitted changes
Best Practices	Use clear branch names like 'feature/login', 'hotfix/api-bug'	Helps team collaboration
Best Practices	Always pull latest develop before branching	Avoids conflicts
Best Practices	Use Pull Requests for merging – code review matters	Ensures quality
Delete the .git folder	rm -rf .git rm -r -fo .git	On macOS/Linux On Windows (Git Bash or PowerShell)

Cấu trúc: <type>(<scope>): <short description>		
Prefix	Ý nghĩa	Ví dụ commit
Add	Thêm mới tính năng, API, test, file	feat(user): add registration endpoint
Cut	Xoá tính năng cũ, file không dùng	chore(payment): remove deprecated payment method
Update	Nâng cấp logic, cải thiện code cho tính năng đang tồn tại	feat(validation): update validation logic for user input
Enhance	Cải tiến tính năng, nâng cấp UX/UI, mở rộng chức năng	feat(search): enhance search feature with keyword highlight
Improve (synonym)	Tối ưu và tăng độ chính xác hoặc hiệu năng tính năng	perf(checkout): improve checkout flow performance
Fix	Sửa bug, lỗi logic, lỗi giao diện	fix(login): correct error message on failure
Refactor	Nếu bạn chỉ tái cấu trúc lại mà không thay đổi hành vi	refactor(report): refactor ReportService to use new utility class
	Áp dụng SOLID	refactor(user): apply Single Responsibility Principle to UserService
	Áp dụng design pattern	refactor(order): implement Strategy pattern for discount calculation
	Tách hàm để KISS	refactor(report): split long method into smaller reusable methods
	Xoá duplicate code (DRY)	refactor(common): extract shared logic into utility class
	Sửa tên biến cho rõ nghĩa	refactor(product): rename ambiguous variables for clarity
Optimize	Nếu bạn cập nhật để tăng hiệu suất hoặc giảm độ phức tạp	perf(image): optimize image resize logic in upload service
Bump	Cập nhật version của package/module	chore(deps): bump spring-boot to 3.1.0
Make	Thay đổi config, Docker, infra	build(docker): switch to multi-stage Docker build
Start	Khởi tạo tính năng mới	feat(dark-mode): initial implementation
Stop	Dừng dùng tính năng hoặc API	refactor(auth): remove legacy email login
Style	Chỉnh format, whitespace, Prettier, xoá comment thừa...	style(codebase): apply Prettier formatting
Document	Cập nhật README, comment code	docs(readme): update API usage instructions
Test	Viết hoặc cập nhật test case	test(report): add unit tests for summary API
Build	Thay đổi hệ thống build (Docker, Webpack, Maven...)	build(maven): add build profile for staging
CI	Cấu hình CI/CD, GitHub Actions, GitLab CI	ci(gitlab): add deploy stage to pipeline
Chore	Việc lặt vặt không ảnh hưởng đến code logic	chore(env): rename .env.example
Revert	Hoàn tác commit trước đó	revert: revert "feat(auth): add login feature"
merge	Dùng khi hợp nhất (merge) một branch vào nhánh chính	merge(feature/login): integrate login module into develop
chore	Dùng khi bạn muốn giữ commit log gọn, không tạo changelog tự động	chore: merge hotfix/validate-email into main
release	Dùng khi merge chuẩn bị release hoặc deploy	release: merge staging into main for deployment