

Git flow — gałęzie, commity i praca zespołowa

 Admin Główny  20.10.2026  45 wyświetleń  0 komentarzy

 Dostępne w językach: 🇵🇱 PL 🇬🇧 EN

[🏠 Start](#) / [Blog](#) / [Programowanie](#)

/ Git flow — gałęzie, commity i praca zespołowa

[← Wróć do bloga](#)

 [Pobierz PDF](#)

Konwencja nazw gałęzi

```
feature/blog-markdown-preview
fix/login-redirect-loop
chore/update-symfony-7.2
release/2.1.0
hotfix/critical-xss-vulnerability
```

Zasady:

- małe litery, słowa oddzielone `-`
- prefix opisuje typ pracy

- krótkie, ale opisowe

Conventional Commits

Format: `typ(zakres): opis`

```
git commit -m "feat(blog): add markdown live preview"
git commit -m "fix(auth): redirect loop on logout"
git commit -m "chore(deps): bump symfony/framework-bundle"
git commit -m "docs(readme): update installation steps"
git commit -m "refactor(api): extract ArticleSerializer"
git commit -m "test(blog): add unit tests for Markdown"
```

Typy: `feat`, `fix`, `chore`, `docs`, `refactor`, `test`, `style`,
`perf`

Typowy workflow

```
# 1. Pobierz aktualizacje z main
git fetch origin
git checkout main
git pull origin main

# 2. Utwórz nową gałąź
git checkout -b feature/blog-search

# 3. Pracuj... commit po commicie
git add src/Controller/BlogController.php
git commit -m "feat(blog): add full-text search engine"

# 4. Przed merge — rebase na main
git fetch origin
git rebase origin/main
```

```
# 5. Push i otwórz Pull Request
git push origin feature/blog-search
```

Interaktywny rebase – porządkuj historię

Przed merge warto posprzątać commity:

```
git rebase -i HEAD~5
```

Możliwe akcje:

- `pick` – zostaw commit
- `squash` / `s` – połącz z poprzednim
- `reword` / `r` – zmień opis commita
- `drop` / `d` – usuń commit

Rozwiązywanie konfliktów

```
git fetch origin
git merge origin/main
```

```
# Git wskaże pliki z konfliktami:
```

```
# CONFLICT (content): Merge conflict in src/Contro
```

```
# Edytuj plik, usuń markery <<< === >>>
```

```
# Następnie:
```

```
git add src/Controller/BlogController.php
git commit
```

.gitignore dla Symfony

```
/vendor/  
/var/  
/.env.local  
/.env.*.local  
/public/bundles/  
/node_modules/  
/.idea/  
/.vscode/  
*.swp
```

Przydatne aliasy

```
git config --global alias.lg "log --oneline --graph  
git config --global alias.st "status -sb"  
git config --global alias.co "checkout"  
git config --global alias.br "branch -vva"
```

Efekt `git lg`:

```
* a1b2c3d (HEAD → feature/blog) feat(blog): add se  
* e4f5g6h (origin/main) fix(auth): token expiry  
*   i7j8k9l Merge pull request #42  
|\n| * l0m1n2o feat(api): add pagination  
* | o3p4q5r chore: update dependencies  
|/
```

🔖 Tagi:

#Linux

#Git

#CI/CD

Udostępnij:

 Facebook

 X

 LinkedIn

 Kopiuj link

Komentarze (0)

Bądź pierwszy – zostaw komentarz!

[Zaloguj się](#), aby dodać komentarz.

FactoryCode

Projekt demonstracyjny zbudowany na Symfony 8, PHP 8.4 i PostgreSQL 16.
Tłumaczenia zarządzane przez panel admina z AI.

NAWIGACJA

 Start

 Blog

 Projekty

 Kontakt

 Polityka prywatności

KONTAKT

 kontakt@factorycode.pl

 GitHub