



# Most Widely Used Conventional Commit Tags

Tag	Purpose / When to Use	Example Commit Message
init	Initial project setup	init: setup Next.js project with Tailwind CSS
setup	Folder structure or environment setup	setup: add folder structure and environment setup
feat	Adding a <b>new feature or functionality</b>	feat: implement navbar and routing
feat(api)	Adding or integrating an <b>API-related feature</b>	feat(api): add user card component with API integration
fix	Fixing a <b>bug or error</b>	fix: resolve API fetch error on 404 status
refactor	Code improvement without changing behavior	refactor: extract reusable button component
style	Formatting or visual changes without logic change	style: adjust padding and colors in header
ui	Improving user interface or layout	ui: improve responsive layout for mobile screens
perf	Optimizing performance	perf: optimize re-renders using React.memo
docs	Documentation or README updates	docs: add project overview and screenshots in README
test	Adding or updating tests	test(user): add unit tests for OTP verification

Tag	Purpose / When to Use	Example Commit Message
chore	Routine maintenance (config, lint, etc.)	chore(env): add JWT secret and expiration to env file
ci	Continuous Integration / Deployment setup	ci: configure GitHub Actions for deployment
build	Build system or dependency changes	build: update dependencies and fix build warnings
deploy	Deployment-related changes	deploy: deploy project on Vercel
revert	Reverting a previous commit	revert: undo navbar state management change
security	Patching or improving security	security: sanitize user input in login form
config	Modifying configuration files	config: update ESLint and Prettier rules

## 💡 Example Professional Commits:

- feat(auth): implement OTP verification and token generation
- fix(login): correct OTP expiry logic
- docs(readme): update API usage instructions
- chore(env): add JWT secret and expiration to env file
- ci: configure GitHub Actions for deployment
- style(auth): format login controller for readability
- test(user): add unit tests for OTP verification

---

## Best Practices to Stand Out

- Use **present tense**: `add` not `added`, `fix` not `fixed`.
- Keep the **subject line under 72 characters**.

- Add **scope** in parentheses if possible: `feat(auth): ...`, `fix(api): ...`.