

RIFERIMENTO RAPIDO CONVENTIONAL COMMITS

Formato messaggi commit, tipi, scope, breaking change

Formato

Struttura del Messaggio di Commit

```
<type>[optional scope]: <description>
[optional body]
[optional footer(s)]
```

Spiegazione delle Parti

type Categoria della modifica (feat, fix, ecc.)
scope Sezione del codice interessata (opzionale)
description Breve riepilogo all'imperativo
body Spiegazione dettagliata (opzionale, dopo riga vuota)
footer Metadati come BREAKING CHANGE o riferimenti issue

Regole

Modo imperativo "add feature" non "added feature"
Tipo in minuscolo feat, non Feat.
Nessun punto La descrizione non termina con "."
Riga vuota Separa body/footer dalla descrizione

Tipi

Tipi Core (rilevanti per SemVer)

feat Nuova funzionalità (attiva bump versione MINOR)
fix Correzione bug (attiva bump versione PATCH)

Tipi Estesi (Convenzione Comune)

build Sistema di build o dipendenze esterne
chore Task di manutenzione (nessuna modifica src/test)
ci Configurazione e script CI
docs Solo modifiche alla documentazione
perf Miglioramento delle performance
refactor Modifica codice che non corregge né aggiunge
revert Annulla un commit precedente
style Formattazione, spazi bianchi (non CSS)
test Aggiunta o correzione di test

Scope

Utilizzo dello Scope

```
feat(auth): add OAuth2 login flow
fix(parser): handle empty input gracefully
docs(readme): update installation steps
refactor(api): extract validation middleware
```

Linee Guida per lo Scope

Nome modulo feat(auth); fix(parser):
Nome layer feat(api); fix(db):
Area funzionale feat(search); fix(checkout):
Dipendenza build(deps); chore(npm):
Ometti se ampio Nessuno scope per modifiche trasversali

Breaking Change

Segnare Breaking Change

```
feat!: remove deprecated login endpoint
feat(api)!: change response format to JSON:API
fix!: drop Node 14 support
```

Breaking Change nello Footer

```
feat(api): change user endpoint response
BREAKING CHANGE: response now returns array
instead of object. Update client parsing.
```

Regole

! dopo type/scope Marcatore abbreviato di breaking change
BREAKING CHANGE: Token footer (sempre maiuscolo)
BREAKING-CHANGE: Anche valido (forma con trattino)
Impatto SemVer Attiva bump versione MAJOR
Qualsiasi tipo I breaking change si applicano a qualsiasi tipo

Esempi

Commit Semplici

```
feat: add email notifications
fix: prevent race condition in checkout
docs: correct typo in contributing guide
style: format with prettier
refactor: simplify error handling logic
```

Con Scope

```
feat(blog): add comment threading
fix(auth): refresh token before expiry
test(api): add missing edge case coverage
ci(github): add Node 20 to test matrix
```

Con Body e Footer

```
fix(parser): handle nested quotes correctly
```

Previously, nested quotes caused the parser to enter an infinite loop. This adds a depth counter to prevent unbounded recursion.

Closes #234

Footer

Token Footer

BREAKING CHANGE: Descrive breaking change all'API
Closes #123 Chiude automaticamente issue al merge
Fixes #456 Chiude automaticamente issue (riferimento fix)

Refs #789 Referenzia issue senza chiuderla
Reviewed-by: nome Attribuzione revisore
Co-authored-by: nome Attribuzione co-autore

Footer Multipli

```
feat(api)!: redesign authentication flow
```

Migrate from session-based to JWT auth.

```
BREAKING CHANGE: /auth endpoints changed
```

```
Closes #101
Refs #98, #99
```

Strumenti

Commit Linting

```
npm install -D @commitlint/cli \
  @commitlint/config-conventional
echo "module.exports = { extends: \
  ['@commitlint/config-conventional'] }" \
  > commitlint.config.js
```

Strumenti Popolari

commitlint Verifica messaggi commit secondo la convenzione
husky Git hook (esegue commitlint al commit)
commitizen (cz) Generatore interattivo messaggi commit
standard-version Changelog automatico + bump versione
semantic-release Pipeline di release completamente automatizzata
release-please Strumento di automazione release di Google

Configurazione Commitizen

```
npm install -D commitizen \
  cz-conventional-changelog
npx commitizen init cz-conventional-changelog
# Usa: npx cz (o git cz con alias)
```

Pattern Comuni

Mappatura Bump Versione

fix: PATCH (1.0.0 → 1.0.1)
feat: MINOR (1.0.0 → 1.1.0)
BREAKING CHANGE MAJOR (1.0.0 → 2.0.0)
docs, style, ecc. Nessun bump versione

Raggruppamento Changelog

```
## [1.2.0] - 2026-03-27
### Features
- add email notifications (abc1234)
### Bug Fixes
- prevent race condition (#123) (def5678)
```

Formato Revert

```
revert: feat(blog): add comment threading
This reverts commit abc1234def5678.
```