

זמן קריאה: 20 שניות

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

Conventional Commits

לכתוב קומיטים
ברמה של
הגדולים





מה זה?

Conventional Commits הוא תקן
מקובל שמסביר איך לכתוב הודעות
Commit בצורה ברורה ואחידה.

ההודעות עוזרות להבין במה שונה כל
שינוי בקוד, אם מדובר בתיקון בעיה,
הוספת פיצ'ר חדש או שינוי משמעותי
שמפר את ההתנהגות הישנה.



מבנה ה-commit

- **סוג:** מציין את מטרת ה-commit (לדוגמה, feat, fix).
- **תחום** (אופציונלי): מבהיר את חלק הקוד שעליו משפיע ה-commit.
- **תיאור:** הסבר תמציתי של השינויים.
- **תיאור מורחב** (אופציונלי): הסבר מורחב של השינויים.
- **כותרות תחתונות** (אופציונליות): מטא-דאטה נוסף, כגון שינויים גרפיים.



```
<type>[optional scope]: <subject>
```

```
[optional description]
```

```
[optional footer(s)]
```



סוגים

הסבר על כל סוג והתרומה שלו לפרויקט:

- **fix** - תיקון באגים בקוד. תיקונים אלה מתורגמים לעדכון PATCH במנגנון Semantic Versioning.
- **feat** - הוספת תכונה חדשה לקוד. תכונות אלו מתורגמות לעדכון MINOR במנגנון Semantic Versioning.
- **build** - שינויים שקשורים למערכת הבנייה או לתלות בתוכנות חיצוניות.
- **chore** - עדכונים שאינם משנים את הקוד או תכונות עיקריות, כמו שינויים בהגדרות.
- **ci** - שינויים בתהליך ה-CI (Continuous Integration) של מערכת הפרויקט.
- **docs** - שינויים או עדכונים של תיעוד בלבד.
- **style** - שינויים שאינם משפיעים על פונקציונליות הקוד, כמו פורמטינג.
- **refactor** - שינוי בקוד שאינו מוסיף תכונה חדשה ואינו מתקן באג.
- **perf** - שיפורים בביצועי הקוד, אשר יכולים להשפיע משמעותית על זמני ריצה ושימוש במשאבים.
- **test** - הוספת בדיקות או שינויים בבדיקות קיימות לוודא תקינות הקוד.
-



הדגשת Breaking Change

ב-**Conventional Commits**, ישנם שני דרכים להצביע על שינוי שובר (Breaking Change). אחת מהם היא על ידי שימוש ב-**!** מיד אחרי ה-type או ה-scope של ההתחייבות. זהו קיצור שמושך תשומת לב לעובדה שהשינוי הוא שינוי קריטי.



feat(api)!: add new endpoint for user authentication














והשנייה היא ב-footer נכתוב הסבר מפורט על השינוי הקריטי.

BREAKING CHANGE: the user authentication API now requires a token in the header instead of the body.

אפשר גם לעשות את שניהם יחד.



דוגמא לפרויקט מסודר

main		97 Branches	1071 Tags	Go to file	t	Add
 alxhub		release: cut the v19.2.0-next.2 release			✓	a7f20eb · last
	.devcontainer	ci: remove circleci as we no longer rely on it (#58615)				
	.github	build: update github/codeql-action action to v3.28.8 (#59830)				
	.husky	build: simplify husky setup (#54315)				
	.ng-dev	ci: configure merge tool to enforce mandatory PR statuses				
	.vscode	refactor(docs-infra): complete removal of aio directory (#59830)				
	.yarn	build: update yarn to 1.22.22 (#58448)				
	adev	fix(docs-infra): reset sandbox runtime on template change				
	contributing-docs	docs: updates versioning docs to recommend that fixes issues				
	devtools	docs: fix spelling of "set up" (#58362)				
	goldens	fix(core): fix race condition in resource() (#59851)				
	integration	feat(core): support TypeScript 5.8 (#59830)				
	modules	docs: update license URL from angular.io to angular.dev				

הדוגמא היא מהקוד בגיט של אנגולר.

אתם מוזמנים לראות בכל פרויקט שמכבד את עצמו את השימוש בקונבציה.

Follow

אהבתם? אל תשכחו
לעקוב לעוד תוכן!



<https://www.linkedin.com/in/omri-rajuan>

