

# חמישה כלים ל־Claude Code: מה להתקין, ולמה

זיכרון בין שיחות, צמצום הקשר, ניתוב למודלים והמלצות לפרויקט. בחרו את התוספת שפותרת את הבעיה שלכם.

2026 באוקטובר · NexAi - Gilad · עודכן ב־1

הגרסה המעודכנת באתר: [nex-ai.co/he/guides/claude-code-tools/](https://nex-ai.co/he/guides/claude-code-tools/)



Claude-Mem מוסיף זיכרון בין שיחות, Headroom מצמצם הקשר שנשלח למודל, OmniRoute מנתב בקשות לספקים שונים, Claude Code Setup מציע התאמות לפרויקט, ו־Task Observer מציע שיפורים ל־Skills מתוך העבודה שלכם. הם פותרים בעיות שונות. לא צריך להתקין את כולם, ולא כולם פלאגינים שמתקינים באותה דרך.

המדריך מיועד למי שכבר עובד עם Claude Code ורוצה לבחור תוספת לפי צורך אמיתי. זה אינו מדריך לתוספים של הצי'אט הרגיל ב־claude.ai. ההוראות מבוססות על תיעוד הפרויקטים שנבדק ב־1 באוקטובר 2026, ולא על מבחן ביצועים שערכנו לכלים.

## איזה כלי מתאים לכם?

| הבעיה                                      | הכלי              | סוג החיבור                       |
|--------------------------------------------|-------------------|----------------------------------|
| לא ברור אילו אוטומציות כדאי להוסיף לפרויקט | Claude Code Setup | פלאגין מהמאגר הרשמי של Anthropic |

| הבעיה                                     | הכלי          | סוג החיבור                         |
|-------------------------------------------|---------------|------------------------------------|
| צריך להסביר שוב החלטות משיחות קודמות      | Claude-Mem    | פלאגין זיכרון עם שירות נלווה       |
| פלטי כלים ולוגים מעמיסים על ההקשר         | Headroom      | במדריך הזה: מעטפת הפעלה ופרוקסי    |
| רוצים לנהל חיבור למספר ספקי מודלים        | OmniRoute     | שרת ניתוב מקומי                    |
| חוזרים על אותם תיקונים ורוצים לשפר Skills | Task Observer | Skill לתיעוד תצפיות ולהצעת שיפורים |

## לפני שמתחילים

ודאו ש־Claude Code עובד בפרויקט ניסוי בלי התוספות. שמרו גיבוי להגדרות ורשמו איך אתם מפעילים אותו כיום. את פקודות ה־plugin/מזינים בתוך שיחה ב־Claude Code; את פקודות npm, uv ושמות הכלים מריצים בטרמינל. בהתקנת פלאגין בדקו מה הוא מוסיף ובאיזה היקף: רק לפרויקט הזה, לכל הפרויקטים שלכם, או להגדרות משותפות לצוות. לניסוי אישי, העדיפו התקנה מקומית לפרויקט אם האפשרות זמינה. [הסבר רשמי על התקנה והיקפי הרשאה](#). אל תתחילו בפרויקט שמכיל סודות או מידע של לקוחות. רישיון קוד פתוח אינו אומר ששימוש במודלים או בשירותי ענן הוא חינומי.

## 1. Claude Code Setup: להתחיל מהצרכים של הפרויקט

זהו פלאגין המלצות של Anthropic. הוא מנתח את בסיס הקוד ומציע התאמות כמו Skills, Hooks ושרתי MCP. לפי התייעוד, פעולת ההמלצה היא לקריאה בלבד: היא לא מיישמת את ההצעות ולא משנה קבצים. [המאגר הרשמי](#). בתוך Claude Code, מתוך תיקיית הפרויקט: אפשר למצוא אותו גם [בעמוד הרשמי של Claude Code Setup](#). זהו אותו כלי, ולא התקנה נוספת.

```
/plugin install claude-code-setup@claude-plugins-official
```

אשרו את היקף ההתקנה ובצעו טעינה מחדש אם הממשק מבקש. לאחר מכן כתבו:

בדוק את הפרויקט והמלץ על שתי אוטומציות שכדאי להוסיף קודם. הסבר איזו בעיה כל אחת פותרת. אל תשנה קבצים בשלב הזה.

**איך בודקים שזה עוזר?** חפשו המלצה שמחוברת לכלים ולפקודות הבדיקה בפרויקט שלכם, ולא רשימת התקנות כללית. בחרו הצעה אחת לבדיקה. בקשה נפרדת ליישום עשויה לשנות קבצים, גם אם שלב ההמלצות עצמו אינו עושה זאת.

## 2. Claude-Mem: לשמור הקשר בין שיחות

Claude-Mem שומר תצפיות מעבודה עם כלים, מסכם אותן ומחזיר הקשר רלוונטי לשיחות הבאות. הוא עשוי להתאים לעבודה מתמשכת שבה חוזרים על החלטות והסברים. זיכרון אוטומטי אינו תחליף לתיעוד מסודר של הפרויקט. [הוראות ומידע במאגר Claude-Mem](#).

בתוך Claude Code:

```
/plugin marketplace add thedotmack/claude-mem
/plugin install claude-mem
```

לאחר ההתקנה הפעילו מחדש את Claude Code והשלימו את בחירת הספק אם תתבקשו. מסלול ההתקנה עשוי להציע התחברות ושירות זיכרון מתארח; בדקו היכן נשמר המידע ואיזה ספק מעבד אותו. אל תניחו שכל התהליך מקומי.

**בדיקת ניסיון:** קבלו החלטה לא רגישה בשיחה אחת, למשל שם לפונקציית עזר. פתחו שיחה חדשה ובדקו אם ההקשר הרלוונטי חוזר. זו הצעת בדיקה, לא הבטחה שכל פרט יישמר או יישלף.

שימו לב: `npm install -g claude-mem` לבדו אינו מתקין את חיבורי הפלאגין והשירות הדרושים. לפני עבודה אמיתית, בדקו גם אפשרויות החרגה ומחיקה של מידע.

## 3. Headroom: לצמצם פלט לפני שהוא מגיע למודל

Headroom נועד לצמצם את ההקשר שמועבר למודל, למשל פלטי כלים ארוכים. כאן נשתמש במסלול מעטפת ההפעלה המתועד שלו. התועלת תלויה בסוג העבודה: אין בסיס להבטיח אחוז חיסכון קבוע או איכות תשובות זהה בכל משימה. [התקנה ומגבלות במאגר Headroom](#).

אם uv כבר מותקן, הריצו בטרמינל:

```
uv tool install --python 3.13 "headroom-ai[all]"
headroom doctor
headroom wrap claude --code-memory none
```

הדגל `code-memory none` מדלג על שילוב Serena, שהמעטפת עשויה להוסיף אחרת להגדרות המשתמש. כך הניסוי נשאר ממוקד בדחיסת ההקשר. צריך להפעיל את Claude דרך המעטפת כדי להשתמש במסלול הזה; התקנת החבילה לבדה אינה מספיקה.

**איך בודקים?** הריצו משימה קטנה עם פלט לוגים ארוך, והשוו להרצה ללא המעטפת. בדקו את התוצאה ואת הבדיקות האוטומטיות, לא רק את מספר הטוקנים. אפשר להיעזר ב־`headroom perf` להצגת נתוני ביצועים.

דחיסה מקומית לא הופכת את המודל למקומי: תוכן עדיין נשלח לספק המודל. במקרה של תקלה, חזרו להפעלה הרגילה והיעזרו בתיעוד לפני שינוי הגדרות נוספות.

## 4. OmniRoute: לנהל את ניתוב הבקשות

OmniRoute הוא שער למודלים, לא פלאגין רגיל של Claude Code. הוא שימושי כשיש צורך בחיבור למספר ספקים או בניהול מסלולי בקשות. אם אתם עובדים עם ספק אחד והכול עובד, ייתכן שאין לכם צורך בשכבה הזאת. [מאגר](#) OmniRoute.

עם Node.js ו-npm בגרסה הנתמכת בפרויקט, הריצו בטרמינל:

```
npm install -g omniroute  
omniroute
```

פתחו את `http://localhost:20128`, השלימו את הגדרת הספק והמודל, ואז השתמשו בהוראות החיבור ל-Claude Code מתוך [מדריך ההגדרה של OmniRoute](#). אל תעתיקו כתובת API אקראית להגדרות קיימות ואל תחליפו קובץ הגדרות שלם.

**בדיקת ניסיון:** שלחו בקשה לא רגישה ובדקו ביומן לאיזה ספק ומודל היא הגיעה. ודאו שמובן לכם מי מחייב על השימוש. שער מקומי עדיין מעביר בקשות לספקים חיצוניים; הוא אינו מעניק Claude ללא הגבלה או מבטל את תנאי הספק.

## 5. Task Observer: להפוך תיקונים להצעות לשיפור

Task Observer של Eoghan Henn / rebelytics מתעד דפוסים ותיקונים במהלך העבודה ומציע לעדכן Skills קיימים או ליצור חדשים. אתם עוברים על ההצעות לפני היישום. בניגוד ל-Claude-Mem, המטרה כאן היא לשפר את הוראות העבודה, ולא רק לשחזר הקשר. [המאגר והוראות ההתקנה](#).

מסלול התקנה מתועד בטרמינל:

```
npx skills add rebelytics/one-skill-to-rule-them-all --skill task-observer
```

בדקו שההתקנה מיועדת ל-Claude Code ובהיקף הרצוי. בהתקנה ידנית שומרים את `SKILL.md`, `references` ו-`scripts` יחד בתיקייה `./claude/skills/task-observer`. בפרויקט.

ההתקנה לבדה אינה מספיקה: עקבו אחר הוראות ההפעלה שבמאגר, באמצעות הנחיה ב-`CLAUDE.md` או Hook לתחילת שיחה, ובדקו בשיחה חדשה שה-Skill מופעל. אין צורך להפעיל מיד תהליך אוטומטי לעדכון Skills.

**בדיקת ניסיון:** תקנו הנחיה לא רגישה במהלך משימה ובקשו לראות את התצפיות שנרשמו. בחנו הצעת שינוי אחת לפני שמאשרים אותה. יומני התצפיות עשויים להכיל פרטי עבודה; בדקו אותם לפני שיתוף או שמירה ב-Git.

## באיזה סדר כדאי לנסות?

ההמלצה שלנו: התחילו מ-Claude Code Setup אם אינכם יודעים מה חסר. אם הבעיה ברורה, עברו ישיר לכלי המתאים: זיכרון, צמצום הקשר או ניתוב. הוסיפו כלי אחד, בדקו משימה מייצגת, ורק אז שקלו תוספת נוספת.

אל תפעילו מיד את Headroom ו־OmniRoute יחד. שניהם יכולים להשפיע על מסלול הבקשות, ותאימות השרשור ביניהם לא נבדקה במסגרת המדריך. שמרו דרך פשוטה לחזור להגדרות המקוריות.

את Task Observer כדאי לשקול כשאתם חוזרים על אותם תיקונים או מנהלים כמה Skills. התחילו מבדיקה ידנית של ההצעות, בלי לאשר מראש שינויים לכל ספריית ה־Skills.

## שאלות נפוצות

### האם מתקינים את כולם דרך אותו תפריט?

לא. Claude-Mem ו־Setup מותקנים כאן דרך מנגנון הפלאגינים. את Headroom ו־OmniRoute מפעילים דרך הטרמינל. Task Observer מותקן כ־Skill ודורש גם הגדרת הפעלה.

### האם הם חינמיים?

אין מחיר אחד לכל המערכת. בדקו את רישיון הכלי, שירותי הענן הנלווים ועלויות ספקי המודלים לפני התחברות או הזנת פרטי תשלום.

### האם צריך את כולם כדי לעבוד עם Claude Code?

לא. תוספת כדאית רק אם היא פותרת בעיה שאתם פוגשים בפועל. התקנה נוספת גם מוסיפה הגדרות, הרשאות ותחזוקה.

### האם אפשר לעבוד עם מידע רגיש?

אל תניחו זאת. בדקו מדיניות שמירה, ספקי עיבוד, הרשאות, לוגים ואפשרויות מחיקה. בצוות, קבלו אישור לפני הוספת שירות שנחשף לקוד או למידע עסקי.

[לכל המדריכים של NexAi](#)

---

לעדכונים, להעתקת פרומפטים בלחיצה ולקישורים נוספים, חזרו למדריך באתר.