Package.json/Packge-Lock.json

10:17 AM Wednesday, August 7, 2024

מה זה בעצם Package.json?

Package.json הוא הלב של Node.js, הוא מכיל בתוכו את המידע בתור MetaData שאותו MetaData יכול להיות מחולק ל-2 קטגוריות

1. **Identifying MetaData Properties:** זהו MetaData שמכיל פרטים מזהים של ה-Module/Project כגון שם הפרויקט, הכותב של הפרויקט, הגרסה שלו, רישיון במידה ויש...
2. **Functional metadata properties:** זהו MetaData המכיל את ה-Values/Properties שנמצאים ב-Module/Project כגון Entry/Starting Point Of The Module, ה-Dependencies שנמצאים בפרויקט, Scripts שבשימוש...

מה ההבדל בין Package.json ל-Package-Lock.json

ראשית נתייחס אל מספרי הגרסאות ב-Package.json.

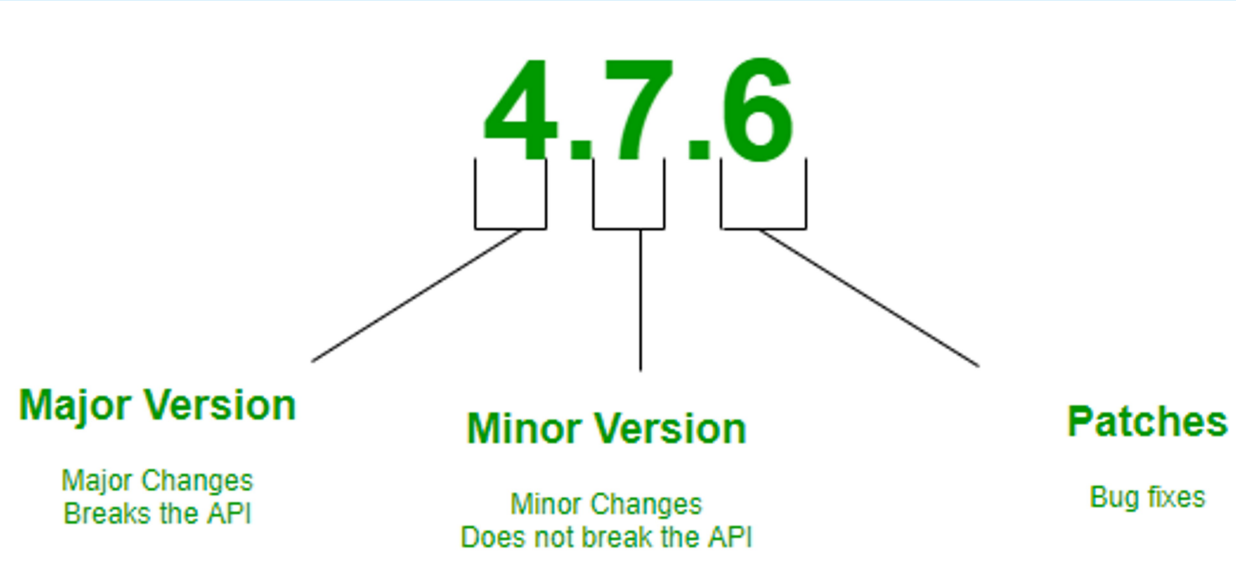
"@nestjs/common":"^10.0.2"

נתייחס אל הדוגמא מעלינו, הגרסאות של Dependency מוצגות בתור 3 ספרות שבין כל ספרה נקודה.

הספרה השמאלית, מראה שינוי משמעותי ב-Dependency ניתן לראות כי ב-@nestjs/common היו כ-10 שינויים משמעותיים בקוד "ששוברים את ה-API" (משמע, שינוי ב-Code-Base של ה-API)

הספרה האמצעית, מראה שינוי קטן כמו הוספה של Feature או שינוי בקוד שאיננו שובר את ה-API, ניתן לראות כי ב-@nestjs/common היו 0 שינויים קטנים.

הספרה הימנית, מראה תיקון באגים, ניתן לראות כי ב-@nestjs/common תיקנו 2 באגים.



עכשיו שהבנו איך עובדים מספורי הגרסאות מה אומר הסימן ^ שליד הגרסה?

ב-Package.json וב-Package-Lock.json יש 2 סוגי סימונים שיכולים להופיע ליד הגרסאות:

1. **^** שמשמעותו, כל פעם שיש שינוי קטן לגרסה הגדולה בה אנחנו נמצאים ותיקוני באגים לאותה הגרסה, תעדכן לגרסה החדשה ביותר ותתקן את הבאגים בצורה אוטומטית. לדוגמא, ב-@nestjs/common אם יש שינוי קטן ותיקון באגים שהיא נגיד הגרסה ה-10.1.3 אז הפרויקט יתעדכן אוטומטית.
2. **~** שמשמעותו, אל תשנה את השינויים הגדולים והקטנים גם אם יש גרסאות חדשות יותר, אבל כל פעם שיש תיקון באגים לגרסה בה אני נמצא, תתקן את הבאגים בצורה אוטומטית. לדוגמא, ב-@nestjs/common אם יש תיקון באגים שהיא נגיד הגרסה ה-10.0.1 אז הפרויקט יתעדכן אוטומטית, לעומת זאת אם ישנו שינוי קטן חדש אז הפרויקט לא יתעדכן.

וכאן הגענו לתפקידו של ה-Package-Lock.json שלו יש 3 תפקידים:

1. Package-Lock.json הוא אינדקס של כל ה-Packages Versions ב-Node_Modules.
2. כאשר אנו מבצעים את הפקודה npm install אנו מתקינים Packages מה-Package-Lock.json.

3. כאשר אנו מבצעים את הפקודה npm update אנחנו מעדכנים את ה-Package.json ולא את ה-Package-Lock.json.

כל אלו גורמים בסופו של דבר לסטנדרטיזציה של החבילות אותם אנו מתקינים, ללא ה-Package-Lock.json כאשר מישהו יעשה Clone ל-Repository שלנו, הוא יתקין את הגרסה החדשה ביותר של ה-Dependencies שבפרויקט שעלולים ליצור בעיות משום שהמפתח לא הריץ את הפרויקט עם הגרסאות של ה-Dependencies הללו.