

קורס NodeJS תשפה

נספח - Node.js + TypeScript Naming Conventions

1. Files and Folders

- **File Names:** Use **kebab-case**, all lowercase.
Example: user-service.ts, auth-controller.ts
- **Folder Names:** Use **kebab-case**, all lowercase.
Example: middlewares/, routes/, services/

2. Variables and Constants

- **Local Variables:** Use **camelCase**.
Example: userId, accessToken
- **Constants:** Use **UPPER_SNAKE_CASE**.
Example: MAX_RETRIES, DEFAULT_TIMEOUT
- Always use **const** for values that don't change.
- Use **let** for variables that are mutable.

3. Functions and Methods

- **Function Names:** Use **camelCase**.
Example: getUserById(), calculateTax()
- **Method Names:** Use **camelCase**.
Example: save(), validateEmail()

4. Classes and Interfaces

- **Class Names:** Use **PascalCase**.
Example: UserService, AuthController
- **Interface Names:** Use **PascalCase**, optionally prefixed with I.
Example: IUser, ICreateUserRequest

5. Types, Enums, and Generics

- **Type Aliases:** Use **PascalCase**.
Example: `type Json = Record<string, any>`
- **Enums:**
 - Use **PascalCase** for the enum name.
Example: `Role`
 - Use **UPPER_SNAKE_CASE** for enum members.
Example: `ADMIN, USER`
- **Generic Type Parameters:** Use a single uppercase letter (T, U, V, etc.).
Example: `function wrapInArray<T>(value: T): T[] {}`

7. Other Best Practices

- **Environment Variables:** Use **UPPER_SNAKE_CASE**.
Example: `DB_HOST, API_KEY`
- **Private Properties:** Prefix with an underscore `_`.
Example: `_isLoggedIn`
- **Error Classes:** Class names should end with `Error`.
Example: `ValidationError, DatabaseError`
- **Promises:** Name promises clearly to indicate their purpose.
Example: `fetchUserPromise`