Чтобы запустить TypeScript с помощью npm, выполните следующие шаги:

**1. Установите TypeScript:**

   Откройте терминал и выполните следующую команду:

      npm install -g typescript @types/node – для импорта модулей

**2. Инициализируйте проект (package.json):**

   В корневой папке вашего проекта выполните:

      npm init –y

**3. Создайте файл конфигурации TypeScript (tsconfig.json):**

   Запустите:

      npm tsc –init

**4. Для запуска ts файлов в node без компиляции в js**

npm I ts-node-dev

**5. Создайте файл TypeScript:**

   ./src/typescripts/index.ts

**6. Настройка scripts в package.json**

«start»: «ts-node-dev –respown ./src/index.ts» - запуск ts файлов в node с перезагрузкой после каждого изменения

«build»: «tsc» - запуск ts с параметрами, описанными в tsconfig.json (компиляция в js)

Вот основные настройки параметров tsconfig.json:

- compilerOptions: Основные параметры компилятора.

  - target: Определяет версию JavaScript, в которую будет скомпилирован код. (например, ES5, ES6, ES2015, esnext – только для новых браузеров).

  - module: Указывает систему модулей (например, commonjs – запуск в node, esnext, es6 – звпуск в браузере, amd).

  - strict: Включает строгий режим (true или false).

  - outDir: Указывает папку для выходных файлов (например, ./dist).

  - rootDir: Определяет корневую папку с исходными файлами.

  - esModuleInterop: Позволяет использовать стандартные модули ECMAScript с CommonJS.

- include / exclude: Определяют, какие файлы и папки включаются или исключаются из компиляции.

- lib: Указание библиотек, которые будут использоваться (например, ["ES6", "DOM"]).

Эти настройки помогают управлять поведением компилятора TypeScript.

{

  "compilerOptions": {

    "target": "ES6", // Целевая версия JavaScript

    "module": "commonjs", // Система модулей

    "strict": true, // Включение строгой проверки

    "esModuleInterop": true, // Совместимость с CommonJS

    "skipLibCheck": true, // Пропустить проверку библиотек

    "forceConsistentCasingInFileNames": true, // Консистентное написание имен файлов

    "outDir": "./dist", // Выходная директория

    "rootDir": "./src", // Корневая папка исходников

    "sourceMap": true // Генерация карт исходников

  },

  "include": ["src/\*\*/\*"], // Включает все файлы в папке src

  "exclude": ["node\_modules", "dist"] // Исключает папки node\_modules и dist

}