

# NodeJS tsx

## Table of Contents

[Fonctionnalités](#)[Mise en route](#)[Conditions préalables](#)[Démarrage rapide](#)[Installation comme dépendance](#)[Installation globale](#)[Utilisation de tsx](#)[Utilisation en ligne de commande](#)[Utilisation dans package.json#scripts](#)

**tsx** signifie **T**ype**S**cript **E**Xecute et c'est une amélioration de Node.js pour exécuter TypeScript.

Il faut considérer **tsx** comme un alias de **node** et l'utiliser de la même manière :

```
node file.js
```

```
tsx file.ts
```

On peut également transmettre des indicateurs Node CLI et des fichiers JS :

```
tsx --env-file=.env ./file.js
```

## Fonctionnalités

- Exécution TypeScript transparente: On peut exécuter du code TypeScript sans se soucier de la configuration. **tsx** exécute le code TypeScript avec des valeurs par défaut modernes et judicieuses, ce qui le rend convivial et particulièrement adapté aux débutants.
- Importations CJS, ESM transparentes: Plus besoin de se demander si un package est **CommonJS** ou **ESM**. On ne rencontrera plus jamais l'erreur `ERR_REQUIRE_ESM` dans **Node**.
- Mode Watch: En bonus DX, tsx est livré avec le mode Watch pour réexécuter les fichiers chaque fois qu'on les enregistre.

## Mise en route

### Conditions préalables

Avant de pouvoir commencer à utiliser **tsx**, il faut s'assurer que **Node.js** est installé. tsx est conçu pour être compatible avec toutes les versions maintenues de **Node.js**.

## Démarrage rapide

**tsx** peut être exécuté avec **npx**, un outil permettant d'exécuter des packages **npm** sans les installer.

Dans la ligne de commande, il suffit d'indiquer le fichier TypeScript que l'on veut exécuter:

```
npx tsx ./script.ts
```

## Installation comme dépendance

Pour installer **tsx**, exécuter l'une des commandes suivantes dans le répertoire de projet :

```
# avec npm

npm install -D tsx

# ou avec pnpm

pnpm add -D tsx

#ou avec yarn

yarn add -D tsx
```

## Installation globale

Pour utiliser tsx n'importe où sur l'ordinateur (sans npx), il faut l'installer globalement :

```
# avec npm

npm install -g tsx

# ou avec pnpm

pnpm add -g tsx
```



**Yarn 2** ne prend pas en charge l'installation globale.

Cela permet d'appeler tsx directement :

```
tsx file.ts
```

## Utilisation de tsx

### Utilisation en ligne de commande

Une fois installé, on peut l'appeler avec le gestionnaire de paquets dans le répertoire du projet :

```
# avec npm

npx tsx ./file.ts

# ou avec pnpm

pnpm tsx ./file.ts

# ou avec yarn

yarn tsx ./file.ts
```

### Utilisation dans package.json#scripts

Les commandes du projet sont généralement organisées dans l'objet **package.json#scripts**.

Dans l'objet scripts, on peut référencer **tsx** directement sans **npx**:

```
// package.json
{
  "scripts": {
    "start": "tsx ./file.ts"
  }
}
```

From:

<http://vps101364.serveur-vps.net/> - dwndoc

Permanent link:

<http://vps101364.serveur-vps.net/doku.php?id=notes:nodejs-tsx>

Last update: **2025/02/19 10:59**

