



# Mises à jour (NodeJS, NPM, Angular et Devextreme)

## NodeJS et NPM

### Mettre à jour Node.js

1. Aller sur le site officiel : <https://nodejs.org/>
2. Télécharger la version **Current**, pas **LTS**).
3. Lancer l'installateur et suivre les instructions.
4. Vérifier la version : `node -v`

### Mettre à jour npm

1. `npm install -g npm@latest`
2. Vérifier la version : `npm -v`

**Pour la librairie, faire cette étape dans chaque lib**

### Nettoyer et réinstaller les dépendances du projet

1. Pour une install clean : `npm ci`
2. Si ça bloque :
  1. Redémarrer son PC (pour qu'il prenne en compte la MàJ de node)
  2. Supprimer le fichier `package-lock.json` et le dossier `node_modules`
  3. Nettoyer le cache npm : `npm cache clean --force`
  4. Réinstaller toutes les dépendances : `npm install`

**Pour la librairie, revenir à la racine**

### Vérifier que tout fonctionne

1. Vérifier la version exacte de Node.js et npm : `node -v` puis `npm -v`
2. Tester la compilation : `.\Compile.bat`

**Pour la librairie, revenir à la racine**

## Angular

- Mettre à jour Angular v18.0 → v19.0 :

1. Mettre à jour Angular : `ng update @angular/core@19 @angular/cli@19`
2. à la question « Use application builder ? » faire Espace puis Entrée
3. à la question « Provide initializer » faire Entrée

- Mettre à jour Angular v19.0 → v20.0 : suivre :

1. Mettre à jour Angular : `ng update @angular/core@19 @angular/cli@19`
2. à la question « Use application builder ? » faire Espace puis Entrée
3. à la question « Provide initializer » faire Entrée
4. mettre à jour `angular/material` : `ng update @angular/material@19`

## TypeScript

- Option 1 : Mettre à jour automatiquement à la dernière version compatible
  1. `ng update typescript` cela mettra à jour TypeScript vers la version recommandée pour ta version actuelle d'Angular.
- Option 2 : Installer une version spécifique (TypeScript 5.8 ou plus pour Angular 20)
  1. `npm install typescript@latest --save-dev`

## Migration du gestionnaire d'erreurs du Router

Remplacer le `main.ts` par :

```
import { bootstrapApplication } from '@angular/platform-browser';
import { provideHttpClient } from '@angular/common/http';
import { provideRouter, withNavigationErrorHandler } from '@angular/router';
import { AppComponent } from './app.component';
import { ThemeService } from './theme.service';
import { Route } from './app.routes';

// Définition de ton gestionnaire d'erreurs personnalisé
function myCustomErrorHandler(error: any) {
  console.error('Navigation error:', error);
}

bootstrapApplication(AppComponent, {
  providers: [
    provideHttpClient(),
    ThemeService,
    provideRouter(Route, withNavigationErrorHandler(myCustomErrorHandler))
  ]
})
  .catch((err) => console.error(err));
```

## Devextreme

1. `npm install devextreme@25.1 --save --save-exact`
2. `npm install devextreme-angular@25.1 --save --save-exact`
3. uniquement pour la lib :
  1. `npm install devextreme-themebuilder@25.1 --save --save-exact`
  2. `npm run build-themes`

## Régler les potentielles erreur

La mise à jour de TypeScript, qui est de moins en moins permissif, entraîne des erreurs lors de la compilation ([voir la documentation](#)).

La mise à jour de Devextreme également, entraîne pour sa part, des erreurs liées à certaines propriétés des composants de sa bibliothèque qui sont désormais dépréciés ou inexistantes.

La mise à jour de Angular, provoque des erreurs liées aux usages des method et variables.

exemples d'erreur remontées :

Erreur n°1	
Erreur	<b><i>Argument of type 'string   number   Date   null' is not assignable to parameter of type 'string   number   Date'. Type 'null' is not assignable to type 'string   number   Date'. [plugin angular-compiler]</i></b>
Explication	Le \$event peut potentiellement être de type null, hors, dans la méthode on n'attend pas ce type d'argument. Il faut ajouter le type null et le gérer.
HTM/TS	<pre>date-box-field.component.html: (valueChange)="handleDateBoxFieldChanged(\$event)"  date-box-field.component.ts: handleDateBoxFieldChanged(itemValue: Date   number   string ) {   this.item.valueString = this.item.formatDateOnly(new Date(itemValue));   this.item.value = this.item.valueString;   this.dateBoxFieldChange.emit(this.item); }</pre>
Correction	<pre>handleDateBoxFieldChanged(itemValue: Date   number   string   null) {   if(!itemValue) {     this.item.valueString = '';   }else {     this.item.valueString = this.item.formatDateOnly(new Date(itemValue));   }   this.item.value = this.item.valueString;   this.dateBoxFieldChange.emit(this.item); }</pre>
Erreur n°2	
Erreur	<b><i>This condition will always return true since this function is always defined. Did you mean to call it instead? [plugin angular-compiler]</i></b>

<b>Erreur n°2</b>	
Explication	Erreur lors de l'appel de la méthode. Oubli des parenthèses
HTML/TS	detail-personne-informations.component.html: <code>@if (DG.isDroitsFichierLectureComptable) {...}</code>
Correction	<code>@if (DG.isDroitsFichierLectureComptable()) {...}</code>
<b>Erreur n°3</b>	
Erreur	<b>Can't bind to 'closeOnOutsideClick' since it isn't a known property of 'dx-popup'.</b> <b>1. If 'dx-popup' is an Angular component and it has 'closeOnOutsideClick' input, then verify that it is included in the '@Component.imports' of this component.</b> <b>2. If 'dx-popup' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@Component.schemas' of this component to suppress this message.</b> <b>3. To allow any property add 'NO_ERRORS_SCHEMA' to the '@Component.schemas' of this component. [plugin angular-compiler]</b>
Explication	closeOnOutsideClick est déprécié depuis la version 24 de Devextreme, mais ne générerait pas d'erreur hideOnOutsideClick remplace désormais cette propriété dans les DxPopup
HTML/TS	logeas-grid.component.html: <code>[closeOnOutsideClick]="false"</code>
Correction	<code>[hideOnOutsideClick]="false"</code>
<b>Erreur n°4</b>	
Erreur	<b>LoadIndicateurModule is not used within the template of FichierMailingComponent</b>
Explication	Tous les imports non utilisés provoqueront un <b>Warning</b> .
Correction	Penser à nettoyer son code et à supprimer les imports qui ne sont plus nécessaires
<b>Erreur n°5</b>	
Erreur	<b>Compiling with Angular sources in partial compilation mode.WARNING: ▲ [WARNING] Deprecation [plugin angular-sass] Global built-in functions are deprecated and will be removed in Dart Sass 3.0.0. Use color.alpha instead. More info and automated migrator</b>
Explication	L'utilisation des variables dans le scss doivent être modifiés pour correspondre aux attentes de SASS qui lancera sa mise à jour prochainement
CSS	<code>color: rgba(\$base-text-color, alpha(\$base-text-color) * 0.7);</code>
Correction	<code>color: rgba(\$base-text-color, color.alpha(\$base-text-color) * 0.7);</code>
<b>Erreur n°5</b>	
Erreur	<b>ompiling with Angular sources in partial compilation mode.WARNING: ▲ [WARNING] Deprecation [plugin angular-sass] Sass @import rules are deprecated and will be removed in Dart Sass 3.0.0. More info and automated migrator</b>
Explication	Les imports dans le scss doivent être modifiés pour correspondre aux attentes de SASS qui lancera sa mise à jour prochainement
CSS	<code>@import "../themes/generated/variables.additional.scss";</code>
Correction	<code>@use "../themes/generated/variables.additional.scss";</code>

From:

<https://wiki-logeas.fr/certif/> - **dokuwiki-certif**

Permanent link:

<https://wiki-logeas.fr/certif/doku.php?id=certif:rgpd:installposte:postedev&rev=1772692803>

Last update: **2026/03/05 07:40**

