Retour au Dossier Organisationnel
Proc#136 - Poste de travail collaborateur : installation et usage du cloud

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 jourangular/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))
// □ Nouvelle méthode
  ]
})
```

```
  .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, entraine des erreurs lors de la compilation (voir la documentation).
La mise à jour de Devextreme également, entraine pour sa part, des erreurs liées à certaines propriétés des composants de sa bibliothèque qui sont désormais dépréciés ou inexistants.
La mise à jour de Angular, provoque des erreurs liées aux usages des method et variables.

exemples d'erreur remontées :

Erreur n°1	
Erreur	***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]***
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	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);`
	`}`

Correction	`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);`
	`}`

Erreur n°2	
Erreur	***This condition will always return true since this function is always defined. Did you mean to call it instead? [plugin angular-compiler]***
Explication	Erreur lors de l'appel de la méthode. Oubli des parenthèses
HTML/TS	detail-personne-informations.component.html: `@if (DG.isDroitsFichierLectureComptable) {...}`
Correction	`@if (DG.isDroitsFichierLectureComptable()) {...}`

Erreur n°3	
Erreur	***Can't bind to 'closeOnOutsideClick' since it isn't a known property of 'dx-popup'. 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. 2. If 'dx-popup' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@Component.schemas' of this component to suppress this message. 3. To allow any property add 'NO_ERRORS_SCHEMA' to the '@Component.schemas' of this component. [plugin angular-compiler]***
Explication	`closeOnOutsideClick` est déprécié depuis la version 24 de Devextreme, mais ne générait pas d'erreur `hideOnOutsideClick` remplace désormais cette propriété dans les DxPopup
HTML/TS	logeas-grid.component.html: `[closeOnOutsideClick]="false"`
Correction	`[hideOnOutsideClick]="false"`
Erreur n°4	
Erreur	***LoadIndicateurModule is not used within the template of FichierMailingComponent***
Explication	Tous les imports non utilisés provoqueront un **Warning**.
Correction	Penser à nettoyer son code et à supprimer les imports qui ne sont plus nécessaires
Erreur n°5	
Erreur	***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***
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	`color: rgba($base-text-color, alpha($base-text-color) * 0.7);`
Correction	`color: rgba($base-text-color, color.alpha($base-text-color) * 0.7);`
Erreur n°5	
Erreur	***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***
Explication	Les imports dans le scss doivent être modifiés pour correspondre aux attentes de SASS qui lancera sa mise à jour prochainement
CSS	`@import “../../themes/generated/variables.additional.scss”;`
Correction	`@use “../../themes/generated/variables.additional.scss”;`

From:
https://wiki-logeas.fr/certif/ - **dokuwiki-certif**

Permanent link:
https://wiki-logeas.fr/certif/doku.php?id=certif:rgpd:installposte:postedev&rev=1772692823

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