Mise en place des dx-validator

Les différents dx-validator

A l'intérieur d'une zone de saisie devetrem (<dx-text-box>, <dx-check-box>, <dx-date-box>...), il est possible de mettre en place des **validateurs de saisie**
il se présente sous la forme

```
  <dx-validator validationGroup="infoBanque">
       <dxi-validation-rule type="custom"
[validationCallback]="customValidator.isValidBicFromDx" message="Le BIC est
invalide"/>
  </dx-validator>
```

- **infoBanque** corresponds au nom d'un groupe de validateur (dans le cas général on peut utiliser le même pour tous ceux d'un composant simple
- **custom** corresponds au type de validation que l'on souhaite (cf plus loin)
    - ici custom = un validateur maison (il est déclaré dans [validationCallback]="customValidator.isValidBicFromDx")

Les différents validateur

Type	Description	Attributs principaux
required	Rend le champ obligatoire.	message
range	Vérifie qu'une valeur est comprise entre deux bornes (dates ou nombres).	min, max, message
stringLength	Contrôle le nombre de caractères.	min, max, trim, message
pattern	Valide la saisie via une expression régulière (Regex).	pattern (ex: /[0-9]/), message
email	Vérifie que le format correspond à une adresse email.	message
compare	Compare la valeur avec un autre champ ou une constante.	comparisonTarget (fonction), comparisonType, message
numeric	Vérifie que la valeur est un nombre.	message
custom	Permet de créer une logique métier complexe en TypeScript.	validationCallback, message, reevaluate
async	Pour les validations nécessitant un appel API (ex: vérification unicité).	validationCallback (doit retourner une Promise/Observable)

Détails techniques des règles

Required (Obligatoire)

C'est le plus commun. Il vérifie que la valeur n'est ni `null`, ni `undefined`, ni une chaîne vide.

```
<dxi-validation-rule type="required" message="Le nom est requis"></dxi-
```

```
validation-rule>
```

Range (Plage de valeurs)

Indispensable pour les dates de naissance ou les montants.

min : Valeur minimum autorisée.

max : Valeur maximum autorisée.

```
<dxi-validation-rule type="range" [max]="today" message="Date
invalide"></dxi-validation-rule>
```

Pattern (Regex)

Idéal pour les codes postaux, numéros de téléphone ou formats spécifiques (IBAN).

```
<dxi-validation-rule type="pattern" pattern="^[0-9]{5}$" message="Code
postal invalide"></dxi-validation-rule>
```

Custom (Personnalisé)

Utilisé quand les règles standards ne suffisent pas. La fonction de callback reçoit un objet contenant la valeur.

```
<dxi-validation-rule type="custom" [validationCallback]="maFonctionTS"
message="Erreur métier"></dxi-validation-rule>
```

Async (Asynchrone)

Utilisé pour vérifier une donnée sur le serveur sans bloquer l'interface.

```
<dxi-validation-rule type="async" [validationCallback]="checkUserExists"
message="Déjà utilisé"></dxi-validation-rule>
```

Propriétés communes à toutes les règles message : Le texte qui sera affiché dans le résumé d'erreurs (votre Toast).

ignoreEmptyValue : (Boolean) Si `true`, la règle n'est pas testée si le champ est vide (sauf pour la règle `required`).

reevaluate : (Boolean) Force la règle à se re-valider même si la valeur n'a pas changé (utile pour les comparaisons entre deux champs).

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

Permanent link:
https://wiki-logeas.fr/certif/doku.php?id=certif:procedure:develop:angular:validator&rev=1773413804

Last update: **2026/03/13 15:56**